EXPLOITING LOW-DIMENSIONAL STRUCTURE AND OPTIMAL TRANSPORT FOR TRACKING AND ALIGNMENT

A Dissertation
Presented to
The Academic Faculty

By

John Z. Y. Lee

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Georgia Institute of Technology

Georgia Institute of Technology

December 2019

EXPLOITING LOW-DIMENSIONAL STRUCTURE AND OPTIMAL TRANSPORT FOR TRACKING AND ALIGNMENT

Approved by:

Dr. Christopher J. Rozell, Advisor
School of ECE
Georgia Institute of Technology

Dr. Justin K. Romberg
School of ECE
Georgia Institute of Technology

Dr. Eva L. Dyer
Dept. of BME, School of ECE
Georgia Institute of Technology

Dr. Mark A. Davenport
School of ECE
Georgia Institute of Technology

Dr. Craig Forest
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: August 1, 2019

Proof itself, of any sort, is impossible, without an axiom (as Gödel proved).

Thus faith in God is a prerequisite for all proof.

Jordan B. Peterson

ACKNOWLEDGEMENTS

I would like to express my gratitude for the generous funding provided by DSO National Laboratories of Singapore so that I could receive my doctoral education at Georgia Tech, where it was a real privilege to work with world class faculty and students. I am immensely thankful to my advisor Chris Rozell, who has been genuinely invested in my growth as a researcher. He not only encouraged my "wild" ideas but, more importantly, taught me how to "reign them in" in a principled way. Most of all, I benefited from his coaching on how to communicate effectively; his unparalleled ability to communicate science will be my personal gold standard for many years to come. I am fortunate to have worked with Eva Dyer who has been extremely supportive of my ideas. From her, I learned the importance of understanding mathematical proofs at an intuitive level, and benefited from her ability to interpret messy application fields with a clear head. Craig Forest shaped my understanding of the relationship between academia and industry through masterful demonstrations of his entrepreneurial tenacity. He showed me that it was possible to take academic research into industry by shrewdly sourcing for well aligned industrial partners. Mark Davenport has always been exceptionally helpful at providing key insights and observations at various points in my research – I have benefited from his vast knowledge and generous spirit. Finally, I am indebted to Justin Romberg's tremendous experience and expertise. He has at many times graciously helped to navigate the complicated path of academia – from illuminating discussions in optimization to advice on how to publish.

I have also had the great privilege of working with exceptional collaborators. I thank Craig Forest and Ilya Kolb for including me in their their brilliant work on automatic patch clampers, Adam Charles and Nick Bertrand for showing me the ropes in modeling and signal processing, and Annabelle Singer, Lu Zhang, Eva Dyer and Max Dabagia for letting me participate in their exciting work in computational neuroscience.

It has truly been a delight working alongside outstanding colleagues, who made my time

at Georgia Tech both meaningful and enjoyable. I thank Adam Charles, Nick (and Muriel), Marissa, Pavel, Adam Willats, Greg, Stephano, Matt, Ayse, Michael, Andy, Liangbei (and Dongxia), Santosh, Andrew, Namrata, Ning, Nishant, Alireza, Karl, and Rakshith for all their simulating discussions and earnest relationships. I will miss the "Dynamics" and "ML" Rozell-lab meetings, where many ideas and papers were spontaneously developed.

I am also very thankful for all the numerous friendships forged outside of work in Atlanta. There is no doubt that these generous individuals have helped to keep me and my family sane: Aster, Denise, Sophia, Ting Ting, Jingfei, Jayden, Mingjuan, Chen, Maggie, Margot, Chih-Li, Angela, Clement, Rachael, Alwynn, Amy, Jerry, Antonella, Ema, Matias, Kyung, Claire, Gowri, Ashish, Sidhu, Sneha, Maria, Guillemo, Clemente, Vanessa, Alethea, Alaina, Gina, Paul, Ian, Maida, Sergio, Diego, Hyunah, Thomas, Henry, Mariko, Isabella, Maria Francisca, Smriti, Liuyu, Anum, Elizabeth, Sevda, Hazel, Nassima, Atefeh, Berni, Griselda, Kiyoe, Ricardo, Teresa, Josh, Rainey, and Father Josh.

With a grateful heart, I thank my close friends and family without whom I would not be where I am today. I thank my parents for being exceptional role models, raising me with a strong faith in God, and teaching me the value of a good education. I thank my godmother Doris and close friend Simon, who have placed tremendous trust in me by being my scholarship guarantors. I thank my in-laws for being incredibly supportive, and being physically present to support my family during our final weeks in Atlanta. I will always be grateful to my family – Teresa, Marie, and Madeleine – for accompanying me on this unforgiving PhD journey; they have no idea just how important they have been and are to me. It is to my incredible wife Teresa that I owe every single one of my PhD successes – this PhD achievement is as much hers as it is mine. Without her patience, sacrifice, faith in me, and tremendous strength in supporting the family, we would not have made it through. To my beautiful daughters Marie and Madeleine, thank you for being my source of joy and purpose – I could have hardly stayed focused without you two. Lastly, I thank God for guiding me though this; I hope whatever contributions made here may glorify Him.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The objective of this thesis is to exploit low-dimensional structures (e.g., sparsity, low-rankness) and optimal transport theory to develop new tools for inference and distribution alignment problems. We investigate properties of structure at two scales: local structure of the single datum along the temporal continuum, and global structure across the dataset's entirety. To study local notions of structure, we consider the fundamental problem of support mismatch under the framework of signal inference: inference suffers when the signal support is poorly estimated. Popular metrics (e.g., ℓ_p-norms) are particularly prone to mismatch due to its lack of machinery to describe geometric correlations between support locations. To fill this gap, we exploit optimal transport theory to propose regularizers that explicitly incorporate geometry. To realize such regularizers at scale, we develop efficient methods to overcome the traditionally-prohibitive computational costs of computing optimal transport. To understand global notions of structure, we consider the challenging problem of distribution alignment, which spans fields of machine learning, computer vision, and graph matching. To bypass the intractability of graph matching approaches, we approach this problem from a machine learning perspective and exploit statistical advantages of optimal transport to align distributions. We develop methods that incorporate manifold and cluster structures that are necessary to regularize against convergence to poor local-minima, and demonstrate the superiority of our method on synthetic and real data. Finally, we present pioneering results in cluster-based *alignability* analysis, which gives us theoretical conditions when datasets can be aligned, as well as error bounds when the alignment transformation is constrained to be isometric.

CHAPTER 1

INTRODUCTION

It is a wonder that patterns exist even in the most chaotic of physical phenomena – almost as though these patterns were encoded by some intelligent design. Following in the footsteps of the great scientific and engineering traditions, we seek to uncover and identify these structures to manipulate them for practical ends. In this thesis, we are interested in the phenomenology surrounding low-dimensional aspects of geometric structure. In particular, we focus on structure at two scales: at the local time-dependent level for structured inference problems, and at the global population level for the structured alignment of multi-modal distributions in machine learning settings.

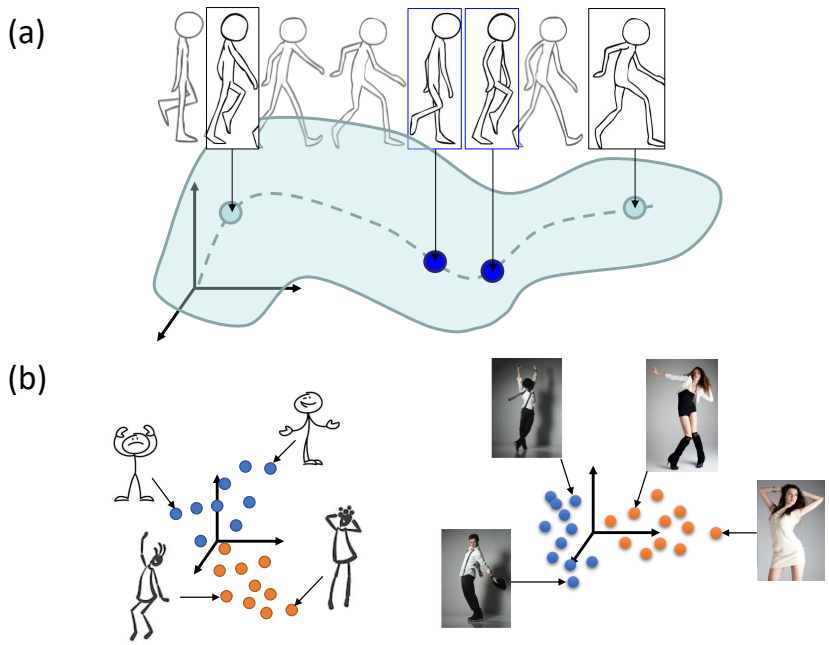


Figure 1.1: In this thesis, we explore two notions of structure: temporal structure and alignable structure. We pose the following fundamental questions: (a) how can we describe structural similarities between datapoints that are close in time, and (b) how can different, but related, datasets be aligned (or mapped), and under what conditions?

1.1 Structured inference

Statistical inference and estimation is a central task in many engineering systems, where we want to accurately determine an underlying signal from noisy and sometimes incomplete sensor measurements. The estimation process can significantly benefit from the incorporation of *a priori* information (structure), because it makes up for information that was corrupted or lost during the sensing process. The development and application of mathematical and statistical notions of *structure* is thus a cornerstone in estimation and inference. Recently, the notion of sparsity-structure has been established as a robust statistical prior for solving numerous machine learning and estimation problems [1, 2, 3]. Sparsity (ℓ_1) regularization has allowed signal-recovery guarantees in challenging scenarios such as compressed sensing [4], yielding remarkable technologies like the single-pixel camera [5].

Temporal dynamical information is another natural source of structure that can be incorporated into the estimation process. A classical approach that optimally[1] propagates dynamical information is the Kalman filter [6]. A crucial point to note is that this tool was designed primarily for a state-space with a Lagrangian (particulated) representation, i.e., each state variable is itself moving through the space (e.g., variables denote displacement coordinates, velocity). The Kalman filter however is not well suited towards signals in a gridded Eulerian representation, where the discretization remain fixed while the signal is travelling across its support (e.g., video). This is in contrast to sparsity approaches, where an Eulerian representation is necessary to describe statistics across its support. To understand why an Eulerian representation suffers in the Kalman framework, one needs to interpret the Kalman filter as a least-squares algorithm: the dynamical penalty is framed as the difference between a signal and its prediction under an ℓ_p loss [7]. From this perspective, the reason why an ℓ_p loss is inadequate emerges: it lacks proper mechanisms to describe geometry between its support locations, and is therefore unable to handle mismatches in signal support. For example, the ℓ_p penalty induced by a support mismatch of 1

[1] When observation noise and dynamical innovations assume Gaussian models.

bin versus 10 bins is equivalent.

Of course, one could work solely in the Lagrangian representation, but doing so sorely neglects powerful tools associated with Eulerian representations (e.g., sparsity based approaches). Hence, for this part of the thesis, we will specifically study methods for signals described by an Eulerian representation. Our approach is to build upon existing sparsity-based approaches and incorporate dynamical information in ways that account for geometry between support locations.

1.2 Structured alignment

Machine learning (ML) is powerful because it performs tasks by automatically learning statistical patterns from the data, relieving the need to manually dictate deterministic programs. Excitingly, ML methods have already reached or exceeded human-performance in visual and audio tasks [8, 9]. Although current ML methods are superb at performing specific tasks, they are not good at generalizing to scenarios where the setting (data or task) has changed only slightly. For example, a system trained to recognize handwriting from one person might not work well on another person's handwriting [10]. More abstractly, a system trained in one domain (features space and data distribution characteristics) might not generalize well to another domain, even when the domains are inherently related. Moreover, in the real world, collection from multiple domains might be difficult and expensive to collect (it could be exhaustive at worst).

A natural question is: how are humans able to generalize across multiple domains so easily? For example, a student reading mathematics is able "port" his framework of logic into the completely different subject of philosophy. The branch in ML that tries to tackle this is known as *transfer learning* (TL), which postulates the following: a learner can draw from a tremendous base of abstract knowledge to effectively transfer prior knowledge to new domains. This spurs another set of questions: *(i)* How do we abstract and represent this base of knowledge? *(ii)* How do we align or register new domains to this base of

knowledge? To study this notion fundamentally, we concern ourselves with *distribution alignment* (DA), which is an unsupervised TL topic. The goal is simple: we wish to *align* two related high-dimensional point-clouds. Though conceptually simple, it surprisingly is still unclear how to do so efficiently and effectively. We hypothesize that introducing *a priori* structure into this problem can help the alignment process. To that end, we propose to study the relatively unexplored idea of alignment based on hierarchical structure under a transfer learning framework.

1.3 Contributions

Because geometric structure is such a universal primitive, many problems may share overlapping structural traits. Consider the challenging scenario of recovering noisy snapshots of time-varying data, sensed through an unknown modality. It is likely that such data lives on a low-dimensional time-varying alignment manifold, and therefore this problem could benefit from combined aspects of inference and alignment. Before tackling such complicated problems however, we first need to better understand basic notions of structure in inference and alignment problems. To that end, we explore three aspects of structure:

- an application of temporal-continuity priors that exploit second order edge-based sparsity statistics;
- the augmentation of inference using optimal transport losses that model the mass transport phenomenon; and
- the alignability of multimodal distributions using hierarchical structures.

In Chapter 3, we design and implement an algorithm that exploits low-dimensional and dynamical structure to improve inference in the application of real-time cell tracking for differential interference contrast microscopy imagery. We adapt existing methods that exploit sparsity and dynamical structure; specifically, we marry dynamical methods that propagate second order statistics (via conjugate priors) with edge-sparsity methods (total

variation (TV) regularization) under a deconvolution framework. We demonstrate the efficacy of a dynamical TV-reweighting method for cell tracking (on synthetic and real data) and integrate a real-time implementation into a robotic patch clamp system for the automation of neuroscience experiments.

In Chapters 4 and 5, we consider various types of optimal transport losses for inference problems. Optimal transport regularizers are interesting because they capture important notions of support uncertainty and deformation. Despite similar ideas being enormously successful in computer vision (i.e., *optical flow* algorithms), optimal transport regularizers have been severely underutilized in the current inference literature due to a few key limitations. We address two critical limitations: (i) strict mass preservation constraints of traditional optimal transport losses are unrealistic in many time-varying physical systems therefore we propose regularizers that relax these; (ii) traditionally exorbitant costs involved with solving optimal transport problems are overcome with our development of fast optimal transport regularizer implementations. Using the proposed optimal transport regularizers in a host of novel dynamical estimation problems, we demonstrate superior reconstruction accuracy and numerical efficiency (in terms of memory and computational speed) with our approach. Chapter 4 develops efficient optimization tools (via proximal methods) for a special case of the optimal transport (1-Wasserstein) regularizer when the signal's support is in a fixed grid with Euclidean spacing, which is a reasonable strategy for imaging applications. Chapter 5 develops a very different set of efficient optimization tools (via Bregman methods) for optimal transport regularizers under general types of support distances, thereby expanding the application domain to a wider range of problems.

In Chapter 6, we consider the unsupervised distribution alignment problem, which is generally known to be NP-hard. When the distributions do not align exactly (e.g., noisy, undersampled) or have complicated multi-modal structure, existing algorithms easily suffer from poor local minima. This necessitates a revisiting of tools that can leverage additional structure in the problem to constrain the solution space. To this end, we propose

a novel method of aligning multimodal distributions by exploiting hierarchical structure. Specifically, we propose a novel nested optimal transport method to account for hierarchical cluster structure, which we efficiently solve using a distributed ADMM manifold algorithm. We study fundamental notions about cluster-based alignability and present conditions when cluster-based alignment is possible as well as performance bounds under an orthonormal alignment strategy. Finally, we demonstrate significant performance improvements over baseline alignment methods on synthetic and real data.

CHAPTER 2

BACKGROUND

2.1 Mathematical preliminaries

2.1.1 Vector and matrix notation

In this thesis, we shall notate vectors and matrices in bold font (e.g., $\boldsymbol{a}, \boldsymbol{A}$) while scalars are notated in standard roman font (e.g., a). Matrices are represented with uppercase letters (e.g., $\boldsymbol{A}$) while vectors are represented with lowercase letters (e.g., $\boldsymbol{a}$). We shall use column-major notation for vectors, where a real-valued vector of n elements is denoted as $\mathbb{R}^n \equiv \mathbb{R}^{n\times 1}$. $\top$ refers to the transpose operator. Scalar subscripts or brackets behind vectors (or matrices) denote elements indices of the vector (or matrix), i.e.,

$$\boldsymbol{a} = [a_1, \ldots, a_n]^\top = [\boldsymbol{a}[1], \ldots, \boldsymbol{a}[n]]^\top \in \mathbb{R}^n,$$
$$\boldsymbol{A} = \begin{bmatrix} A_{11} & A_{21} \\ A_{12} & A_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}[1,1] & \boldsymbol{A}[2,1] \\ \boldsymbol{A}[1,2] & \boldsymbol{A}[2,2] \end{bmatrix} \in \mathbb{R}^{2\times 2}.$$

Inner products between vectors are understood as the standard vector inner product, i.e., $\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \boldsymbol{a}^\top \boldsymbol{b} = \sum_i a_i b_i$. Inner products between matrices are understood as the Frobenius inner product, i.e., $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \sum_{ij} A_{ij} B_{ij}$. When two matrices (or vectors) $\boldsymbol{A}, \boldsymbol{B}$ are of similar size, $\boldsymbol{A} \odot \boldsymbol{B}$ represents elementwise multiplication, while $\boldsymbol{A} \oslash \boldsymbol{B}$ represents elementwise division. Given the vector $\boldsymbol{a}$, $\mathrm{diag}(\boldsymbol{a})$ is the diagonal matrix with diagonal $\boldsymbol{a}$. We denote $\mathbb{1}$ as a column vector of ones (where its size implied by its context). $[\![n]\!]$ refers to the set of indices $\{1, \ldots, n\}$.

2.1.2 Functions

Operators such as $\exp$, $\log$, or the Lambert W function ω, whether applied on vectors or matrices, are always applied elementwise.

We define $\iota_{\mathcal{C}}(\boldsymbol{a})$ to be the indicator function of the set $\mathcal{C}$, i.e., $\iota_{\mathcal{C}}(\boldsymbol{a}) = 0$ if $\boldsymbol{a} \in \mathcal{C}$ and $\iota_{\mathcal{C}}(\boldsymbol{a}) = +\infty$ otherwise. Indicator functions are concatenated as follows: $\iota_{\mathcal{C}}(\boldsymbol{a}, \boldsymbol{b}) = \iota_{\mathcal{C}}(\boldsymbol{a}) + \iota_{\mathcal{C}}(\boldsymbol{b})$. ι_{+} refers to an indicator on the non-negative orthant.

We define the KL divergence on both vectors and couplings as

$$\mathrm{KL}(\boldsymbol{A}|\boldsymbol{B}) \equiv \mathrm{KL}(\boldsymbol{a}|\boldsymbol{b}) \stackrel{\text{def.}}{=} \sum_{i=[\![q^2]\!]} a_i \log(a_i/b_i) - a_i + b_i,$$

$\forall(\boldsymbol{A}, \boldsymbol{B}) \in \mathbb{R}_{+}^{q\times q} \times \mathbb{R}_{+}^{q\times q}$, where $\mathrm{vec}(\boldsymbol{A}) = \boldsymbol{a}$ vectorizes the matrix $\boldsymbol{A}$, and accordingly $\boldsymbol{b} = \mathrm{vec}(\boldsymbol{B})$.

2.2 Maximum a posteriori signal estimation

We shall consider a time-varying linear observation model, whose underlying signal of interest $\boldsymbol{x}_k \in \mathbb{R}^n$ comes from a noisy measurement process

$$\boldsymbol{y}_k = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{\eta}_k, \tag{2.1}$$

with the subscript k denoting a discrete index in time. Here, observations are denoted by $\boldsymbol{y}_k \in \mathbb{R}^m$, and the system of sensors is denoted by $\boldsymbol{A}_k \in \mathbb{R}^{m\times n}$. We also describe an additive noise component with $\boldsymbol{\eta}_k \in \mathbb{R}^m$. To study system dynamics, we shall consider a first-order dynamical Markov model which propagates a signal through time with the function $f_k : \mathbb{R}^n \mapsto \mathbb{R}^n$ as

$$\boldsymbol{x}_k = f_k(\boldsymbol{x}_{k-1}) + \boldsymbol{\nu}_k, \tag{2.2}$$

where $\boldsymbol{\nu}_k \in \mathbb{R}^n$ refers to *innovations* of the dynamical system (e.g., interference or spurious signals acting on the dynamical system).

In many statistical and engineering settings, *maximum a posteriori* (MAP) estimation is known to be one of the most flexible and robust frameworks for framing estimation problems. In this work, we shall adopt this philosophy to motivate the construction of our dynamical framework for estimation. Denoting $\widehat{\boldsymbol{x}}_{k-1}$ as the signal estimate from the previous timestep, we shall assume that $\boldsymbol{x}_k, \widehat{\boldsymbol{x}}_{k-1}, \boldsymbol{y}_k$ are random variables with a joint probability density $p(\boldsymbol{x}_k, \widehat{\boldsymbol{x}}_{k-1}, \boldsymbol{y}_k)$. In addition, we denote the prior density as $p_{\boldsymbol{x}_k}(\boldsymbol{x}_k)$, and the posterior density of $\boldsymbol{x}_k$ conditioned on $\boldsymbol{y}_k$ and $\widehat{\boldsymbol{x}}_{k-1}$ as $p_{\boldsymbol{x}_k|\boldsymbol{y}_k,\widehat{\boldsymbol{x}}_{k-1}}(\boldsymbol{x}_k, \widehat{\boldsymbol{x}}_{k-1}, \boldsymbol{y}_k)$. The MAP framework maximizes the posterior probability as

$$\begin{aligned}
\widehat{\boldsymbol{x}}_k &= \arg\max_{\boldsymbol{x}_k} p_{\boldsymbol{x}_k|\boldsymbol{y}_k,\widehat{\boldsymbol{x}}_{k-1}}(\boldsymbol{x}_k, \widehat{\boldsymbol{x}}_{k-1}, \boldsymbol{y}_k) \\
&\overset{\text{(a)}}{=} \arg\max_{\boldsymbol{x}_k} p_{\boldsymbol{y}_k|\boldsymbol{x}_k}(\boldsymbol{x}_k, \boldsymbol{y}_k) p_{\boldsymbol{x}_k}(\boldsymbol{x}_k) p_{\boldsymbol{x}_k|\widehat{\boldsymbol{x}}_{k-1}}(\boldsymbol{x}_k, \widehat{\boldsymbol{x}}_{k-1}) \\
&\overset{\text{(b)}}{=} \arg\max_{\boldsymbol{x}_k} \Big(\underbrace{\log p_{\boldsymbol{y}_k|\boldsymbol{x}_k}(\boldsymbol{x}_k, \boldsymbol{y}_k)}_{\text{Observation noise}} + \underbrace{\log p_{\boldsymbol{x}_k}(\boldsymbol{x}_k)}_{\text{Signal prior}} + \underbrace{\log p_{\boldsymbol{x}_k|\widehat{\boldsymbol{x}}_k}(\boldsymbol{x}_k, \widehat{\boldsymbol{x}}_{k-1})}_{\text{Dynamical innovations}} \Big),
\end{aligned} \tag{2.3}$$

where (a) is due to Bayes rule, and (b) is due to the monotonicity of the logarithm function. The resulting optimization yields three separate components that contribute to the estimation of $\boldsymbol{x}_k$. The first component is equivalent to a log-likelihood maximization that seeks the signal that maximizes the probability of the signal given measurements; i.e., this captures the statistics of *observation noise*. The second component encourages the signal that best fits the *prior density*; i.e., this captures the statistics of the signal model. The third component is another log-likelihood maximization which finds the signal that maximizes the probability of the signal given its previous dynamical estimate; i.e., this captures the statistics of *dynamical innovations*. This framework is favorable because it is statistically motivated, and its separable-form facilitates the rapid prototyping of new methodologies surrounding each of the separable components.

2.2.1 Sparsity-based convex relaxations

Consider the recovery problem of a k-sparse signal, we denote by $\boldsymbol{x} \in \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_0 \leq k\}$. Let observations $\boldsymbol{y} \in \mathbb{R}^m$ be generated via $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is a sensing (observation) matrix. In the noiseless case, signal recovery may be formulated as the constrained optimization whose objective is exactly the signal prior

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} p_{\boldsymbol{x}_k}(\boldsymbol{x}_k) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}.$$

Since the signal is known to be sparse, a sensible prior is the cardinality of $\boldsymbol{x}$, i.e., $\|\boldsymbol{x}\|_0$:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{x}\|_0 \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}.$$

This problem unfortunately is NP-hard [11], meaning it cannot be solved in polynomial time by any algorithm, hence approximation approaches are the typical way about solving them. The two main approximation approaches are *heuristic-based searches* (e.g., orthogonal matching pursuit [12]) and *convex relaxations*. Approaches using a heuristic-based search are greedy and extremely efficient but they have no global recovery guarantees. On the other hand, convex relaxations are tractable convex surrogate problems that can be provably solved to the global minima. Of convex approaches, the family of ℓ_p norms with $0 < p \leq 1$ are sparsity inducing, with $p = 1$ being most useful because it happens to also be convex. In somewhat of an interesting coincidence, if the Bayesian prior is Laplace distributed (i.e., zero centered with heavy tails), its log-prior is the ℓ_1 norm since $p_{\boldsymbol{x}}(\boldsymbol{x}) \propto \exp(-\lambda\|\boldsymbol{x}\|_1)$. In this thesis, we are primarily interested in the convex relaxation

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{x}\|_1 \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}.$$

The noisy version of this formulation allows discrepancy between the linear measurements and the observations, by replacing the the equality constraint with the inequality

$\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2 \leq \epsilon$, yielding the program (also known as *basis pursuit de-noising* (BPDN) [1] in the signal processing community or the *least absolute shrinkage and selection operator* (LASSO) [3] in the statistical community):

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{x}\|_1 \quad \text{s.t.} \quad \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2 \leq \epsilon$$

or written in Lagrangian form with a user-defined parameter $\lambda > 0$:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 .$$

2.2.2 Reweighted ℓ_1 approaches

The reweighted ℓ_1 framework introduced by Candes *et al.* [13] was found to produce robust reconstructions because each signal element's statistics were individually parameterized as opposed to having them globally parameterized by a single term (in the non-weighted ℓ_1 setup). Denoting t as the algorithmic iteration index, the re-weighted ℓ_1 program is

$$\begin{aligned} \widehat{\boldsymbol{x}}^{(t)} &\leftarrow \arg\min_{\boldsymbol{x}} \ \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda \left\|\boldsymbol{\Lambda}^{(t-1)}\boldsymbol{x}\right\|_1 , \\ \boldsymbol{\Lambda}^{(t)}[i,i] &\leftarrow \frac{1}{|\widehat{\boldsymbol{x}}^{(t)}[i]| + \epsilon}, \qquad \forall i = 1, \ldots, n, \end{aligned}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix, and $\epsilon > 0$ is a small constant that prevents division by zero. The first iteration begins with solving the standard BPDN problem, i.e., $\boldsymbol{\Lambda}^{(0)} = \boldsymbol{I}_n$.

Garrigues and Olshausen [14] furthered this work by casting it in the language of Bayesian inference. They formulated the reweighted ℓ_1 problem as a hierarchical probabilistic model by treating the signal elements as Laplacian random variables conditioned by the weights. A key contribution was to show that the weights themselves could be treated as random variables with Gamma hyperpriors, we denote with $\boldsymbol{\beta}$, and that solving the maximum *a posteriori* estimation using the expectation maximization (EM) approach

admitted a closed form expectation-step for the weight update

$$\boldsymbol{\beta}^{(t+1)}[i] \leftarrow \frac{\tau}{|\widehat{\boldsymbol{x}}^{(t)}[i]| + \eta},$$

where τ, η are positive constants.

In a further development, Charles *et al.* [15] applied this hierarchical Bayesian framework to the dynamical setting. Their key contribution was to design a dynamical ℓ_1 reweighing algorithm by crafting how dynamical information should propagate through the distribution of the hyperprior:

$$\boldsymbol{\beta}_k^{(t+1)}[i] \leftarrow \frac{\tau}{\delta|\widehat{\boldsymbol{x}}_k^{(t)}[i]| + |\widetilde{\boldsymbol{x}}_k[i]| + \eta},$$

where $\widetilde{\boldsymbol{x}}_k = f_k(\widehat{\boldsymbol{x}}_{k-1})$ is understood as a prediction term, and δ, τ, η are positive constants. In essence, a reweighing of the ℓ_1 term using dynamics, second order statistics are propagated through time in a spirit similar to the Kalman filter.

2.2.3 The total variation norm

In computational imaging, a popular sparsity inducing norm that promotes *edge sparsity* (or piece-wise smoothness) is the total-variation norm [16]:

$$\|\boldsymbol{x}\|_{\text{TV}} \stackrel{\text{def.}}{=} \sum_i \|\boldsymbol{T}_i \boldsymbol{x}\|_p \,,$$

where $\boldsymbol{x}$ is understood here as the vectorized version of an image, and $\boldsymbol{T}_i \in \mathbb{R}^{2\times n}$ refers to a linear operator that extracts the first order[1] spatial gradient field of the i-th pixel, i.e., the horizontal and vertical forward-differences. The *anisotropic* TV-norm (when $p = 1$) penalizes spatial differences equally with respect to the ℓ_1 penalty, while the *isotropic* TV-norm (when $p = 2$) penalizes spatial differences with *group sparsity* (which involves taking the

[1] Higher order spatial differences have been shown to be more effective in certain imaging scenarios [17].

ℓ_2 norm over both spatial derivatives). The anisotropic TV-norm is known to produce spatially zig-zagging artifacts especially around transition contours compared to smoother contours of the isotropic TV-norm [18]. The reweighted TV minimization was introduced in [13] as a regularization method (in compressive sensing), for recovering the Shepp-Logan phantom [19] from sparse Radon projections, demonstrating potential for application in bio-imagery.

2.3 Numerical first order convex optimization

In the previous section, we presented statistical frameworks to formulate estimation and inference problems in principled ways. When problems have formulations that abide by convex constraints, problems are guaranteed a global solution regardless of initialization, and a wide selection of reliable numerical methods become available to our disposal. This section introduces relevant background on numerical convex optimization that pertain specifically to this thesis. For comprehensive expositions on the topic of convex optimization, we refer the reader to books by Boyd and Vandenberghe [20] for general convex methods and numerous examples, and by Bauschke and Combettes [21] for convex analysis of monotone operators.

2.3.1 Basic definitions in convex analysis

Before we begin, we present some basic definitions in convex analysis.

Definition 2.3.1 (Domain of a function)**.** *The domain of a function is denoted by*

$$\operatorname{dom} f = \{\boldsymbol{x} \in \mathbb{R}^n : f(\boldsymbol{x}) < +\infty\}.$$

Definition 2.3.2 (Closed function)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is closed if for each $\alpha \in \mathbb{R}$, the sublevel set $\{\boldsymbol{x} \in \operatorname{dom} f : f(\boldsymbol{x}) \leq \alpha\}$ is a closed set.*

Definition 2.3.3 (Proper function)**.** *A function is proper if* $f(\boldsymbol{x}) < +\infty$ *for at least one* $\boldsymbol{x} \in \operatorname{dom} f$ *and* $f(\boldsymbol{x}) > -\infty$ *for all* $\boldsymbol{x} \in \operatorname{dom} f$*.*

Definition 2.3.4 (Convex set)**.** *A set* $\mathcal{C}$ *is convex if the line segment between any two points in* $\mathcal{C}$ *lies in* $\mathcal{C}$*; for any* $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C}$ *and any* $\theta \in [0, 1]$*, we have*

$$\theta \boldsymbol{x}_1 + (1 - \theta)\boldsymbol{x}_2 \in \mathcal{C}.$$

Definition 2.3.5 (Convex function)**.** *A function* $f : \mathbb{R}^n \to \mathbb{R}$ *is convex if* $\operatorname{dom} f$ *is a convex set and if for all* $\boldsymbol{x}$*,* $\boldsymbol{y} \in \operatorname{dom} f$*, and* $\theta \in [0, 1]$*, we have*

$$f(\theta \boldsymbol{x} + (1 - \theta)\boldsymbol{y}) \leq \theta f(\boldsymbol{x}) + (1 - \theta) f(\boldsymbol{y}).$$

Definition 2.3.6 (Strongly convex function)**.** *A differentiable function* f *is* α*-strongly convex if for* $\alpha > 0$*, and for all* $\boldsymbol{x}, \boldsymbol{y} \in \operatorname{dom} f$*,*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\alpha}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 .$$

2.3.2 First order convex optimization

The mature field of iterative methods in convex optimization is "nearly a technology" (quoting Wotao Yin), yet the past few decades have seen a revival of interest in first order approaches due to the explosion of large scale problems (e.g., in computer vision and machine learning). When tackling the issue of scale, considerations such as accuracy, per-iteration costs, memory consumption, and convergence rate need to be carefully balanced. Broadly speaking, first and second order methods represent a major dichotomy in convex optimization methods, where the main trade off is between the rate of convergence (for which second order methods are superior) and the per-iteration computational complexity (for which first order methods are superior). Second order methods, though often convergent within tens of iterations, do not generally scale well due to the complexity associated with the

computation of the second order derivative. It is quite common for second order methods to be excluded due to memory limitations (from the demands of storing and manipulating second order derivatives).

Large scale optimization problems are therefore commonly solved using proximal first order techniques because these methods possess extremely efficient per-iteration costs associated with computing the first order derivative (often parallelizable) at the expense of a slower convergence rate (typically linear). Popular proximal first order approaches include alternating direction method of multipliers (ADMM) [22], the Douglas-Rachford algorithm [23], and primal-dual methods [24]. *Variable splitting* [23, 25] and the *proximal point algorithm* [26] are fundamental building blocks that underlie proximal first-order methods, where the idea is to divide a problem, $f(\boldsymbol{x}) = \sum_i f_i(\boldsymbol{x}_i)$, into a series of easier subproblems that iteratively converge to the global fixed point. These subproblems f_i are solved with *proximal algorithms*, which are defined as

$$\text{Prox}_{\rho f_i}(\boldsymbol{z}) = \underset{\boldsymbol{z}'}{\arg\min} \ \left\{ f_i(\boldsymbol{z}') + \frac{1}{2\rho} \|\boldsymbol{z} - \boldsymbol{z}'\|_2^2 \right\},$$

where f_i is any proper, convex, closed function, and $\rho > 0$ is interpreted as a scalar step size. Due to the iterative nature of such methods, it is of paramount interest for the proximal algorithms of each subproblem to be efficient (e.g., have closed form solutions, or/and have separable form that allows it to be solved with distributed hardware such as *general purpose graphic processing units* – GPGPUs). We refer the reader to [27] for an excellent introductory monograph to the topic.

2.3.3 Primal-dual (Chambolle-Pock) method

Chambolle and Pock [24] introduced a first order *primal dual* method that was provably optimal with a convergence rate of $O(1/n)$. This method is attractive because it is simple to implement, is efficient and easily parallelizable (e.g., when applied to imaging problems).

At the center of our discussion is the prototypical problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x}),$$

where f, g are closed and convex, and $\mathcal{X}$ is a convex set. To motivate the primal dual approach, consider a natural way of attacking this problem by formulating the (Fenchel) dual and applying an iterative proximal method, i.e.,

$$\max_{\boldsymbol{y}} \min_{\boldsymbol{x},\boldsymbol{z}} f(\boldsymbol{x}) + g(\boldsymbol{z}) + \langle \boldsymbol{y}, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{z} \rangle \qquad \Leftrightarrow \qquad \max_{\boldsymbol{y}} -f^*(-\boldsymbol{A}^\top \boldsymbol{y}) - g^*(\boldsymbol{y}),$$

where f^*, g^* are the Fenchel conjugates of f, g respectively. Such a formulation can be directly solved using a proximal gradient algorithm, i.e.,

$$\boldsymbol{y}^{(t+1)} \leftarrow \text{Prox}_{\rho h^*}(\boldsymbol{y}^{(t)} + \rho \boldsymbol{A} \nabla f^*(-\boldsymbol{A}^\top \boldsymbol{y}^{(t)})),$$

where $\rho > 0$ is a step size, but it requires f^* to be smooth (i.e., f strongly convex), which is a limiting constraint (e.g., it excludes functions like $f = \|\boldsymbol{x}\|_1$). The primal dual approach addresses this limitation by formulating the problem as the following saddle point problem

$$\min_{\boldsymbol{x}} \max_{\boldsymbol{y}} -f^*(\boldsymbol{y}) + \langle \boldsymbol{y}, \boldsymbol{A}\boldsymbol{x} \rangle + g(\boldsymbol{x}).$$

As its name suggests, this is solved using proximal gradient descent in $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$\begin{aligned}
\boldsymbol{y}^{(t+1)} &\leftarrow \text{Prox}_{f^*}(\boldsymbol{y}^{(t)} + \rho \boldsymbol{A} \bar{\boldsymbol{x}}^{(t)}) && \text{(dual proximal)} \\
\boldsymbol{x}^{(t+1)} &\leftarrow \text{Prox}_{g}(\boldsymbol{x}^{(t)} - \tau \boldsymbol{A}^\top \boldsymbol{y}^{(t+1)}) && \text{(primal proximal)} \\
\bar{\boldsymbol{x}}^{(t+1)} &\leftarrow \boldsymbol{x}^{(t+1)} + \theta(\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}) && \text{(over-relaxation)}
\end{aligned}$$

where steps sizes $\rho > 0, \tau > 0$ are chosen such that $\rho\tau \|\boldsymbol{A}\|^2 < 1$, and the over-relaxation parameter is $\theta \in [0, 1]$. The convergence rate of this algorithm was shown to be at least

$O(1/n)$ for non-smooth problems, but can be much faster for smoother problems.

2.3.4 Alternating directions method of multipliers (ADMM)

The alternating directions method of multipliers (ADMM) [28] was originally proposed as the Douglas-Rachford algorithm [29] but has recently gained a lot of interest as a simple first order method for distributed applications [22]. ADMM is a versatile framework that effectively applies a divide and conquer strategy: a complicated non-smooth objective is split into separate, simpler ones that can be tackled efficiently. Unlike primal dual approaches, ADMM has a more straightforward formulation that does not involve the Fenchel dual; hence, ADMM is an extremely popular choice when rapid-prototyping. Our problem is

$$\min_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{z}\in\mathcal{Z}} \; f(\boldsymbol{x})+g(\boldsymbol{z}) \qquad \text{s.t.} \qquad \boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{z}=\boldsymbol{b},$$

where f and g are non-smooth convex functions, and $\mathcal{X}, \mathcal{Z}$ are convex sets. The optimization has the following *augmented* Lagrangian

$$\mathcal{L}_\rho(\boldsymbol{x},\boldsymbol{z},\boldsymbol{y}) = f(\boldsymbol{x})+g(\boldsymbol{z})+\iota_{\mathcal{X}}(\boldsymbol{x})+\iota_{\mathcal{Z}}(\boldsymbol{z})+\langle \boldsymbol{y}, \boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{z}-\boldsymbol{b}\rangle+\frac{\rho}{2}\left\|\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{z}-\boldsymbol{b}\right\|_2^2,$$

where $\boldsymbol{y}$ is a Lagrange multiplier, and the equality constraint is relaxed with a quadratic term (weighed by $\rho > 0$) which vanishes when the equality constraint is satisfied. Assuming that the problem is closed, proper, and convex, and there exists a solution to the problem, the ADMM algorithm generates a provably convergent sequence of iterations

$$\boldsymbol{x}^{(t+1)} \leftarrow \operatorname*{arg\,min}_{\boldsymbol{x}\in\mathcal{X}} \; f(\boldsymbol{x})+\langle \boldsymbol{y}^{(t)}, \boldsymbol{A}\boldsymbol{x}\rangle+\frac{\rho}{2}\left\|\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{z}^{(t)}-\boldsymbol{b}\right\|_2^2 \qquad \text{(primal proximal)}$$

$$\boldsymbol{z}^{(t+1)} \leftarrow \operatorname*{arg\,min}_{\boldsymbol{z}\in\mathcal{Z}} \; g(\boldsymbol{z})+\langle \boldsymbol{y}^{(t)}, \boldsymbol{B}\boldsymbol{z}\rangle+\frac{\rho}{2}\left\|\boldsymbol{A}\boldsymbol{x}^{(t+1)}+\boldsymbol{B}\boldsymbol{z}-\boldsymbol{b}\right\|_2^2 \qquad \text{(auxiliary proximal)}$$

$$\boldsymbol{y}^{(t+1)} \leftarrow \boldsymbol{y}^{(t)}+\rho(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{z}-\boldsymbol{b}). \qquad \text{(dual ascent)}$$

Here $\rho > 0$ is understood as a step size, and the individual update steps for $\boldsymbol{x}$ and $\boldsymbol{z}$ are indeed proximal algorithms w.r.t. f, g, respectively. Generally, the ADMM algorithm is known to have a convergence rate of $O(1/n)$ [30], therefore it is efficient when the update steps are themselves efficient (e.g., have a computational complexity that scales linearly). The ADMM is perhaps most effective in distributed settings where the main problem consists of a separable sum where the parts interact minimally – for example, in the total variation denoising imaging problem, the spatial derivative acts on each pixel independently and pixels do not interact among themselves [31]. The ADMM framework has also been shown to converge in several non-convex optimization settings (e.g., under compact manifold constraints like Stiefel) [32] under surprisingly general assumptions.

2.3.5 Bregman ADMM

With ADMM, the splitting introduces a squared ℓ_2 term, to relax the equality constraints. As exposed by the Bregman literature [33, 34, 35], the implicit gradient step induced by the squared ℓ_2-norm may not be optimal w.r.t. the geometry of the set. To overcome this issue, Wang and Banerjee [36] proposed a practical generalization, known as the *Bregman ADMM* (BADMM) to extend Bregman methods to the ADMM framework. In this generalization, the consensus term can be replaced by a Bregman divergence $\mathrm{B}_\phi(\cdot|\cdot)$ that is α-strongly convex. Let $\phi : \Omega \mapsto \mathbb{R}$ be a continuously differentiable and strictly convex function on the relative interior of a convex set Ω, and $\nabla\phi(\boldsymbol{y})$ be the gradient of ϕ at $\boldsymbol{y}$. A classical definition of the Bregman divergence is $\mathrm{B}_\phi(\boldsymbol{x}, \boldsymbol{y}) \overset{\text{def.}}{=} \phi(\boldsymbol{x}) - \phi(\boldsymbol{y}) - \langle \nabla\phi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle$. As an example, when $\phi = \|\boldsymbol{x}\|_2$, $\mathrm{B}_\phi(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$ is the regular squared ℓ_2 distance.

BADMM sets up the prototype problem in a similar way as ADMM:

$$\min_{\boldsymbol{x}\in\mathcal{X}, \boldsymbol{z}\in\mathcal{Z}} f(\boldsymbol{x}) + g(\boldsymbol{z}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{c}, \tag{2.4}$$

where f and g are non-smooth convex functions, and $\mathcal{X}, \mathcal{Z}$ are convex sets. The BADMM

update steps are:

$$\boldsymbol{x}^{(t+1)} \leftarrow \arg\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}) + \langle \boldsymbol{\lambda}^{(t)}, \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z}^{(t)} - \boldsymbol{c}\rangle + \rho \mathrm{B}_{\phi}(\boldsymbol{c} - \boldsymbol{A}\boldsymbol{x}, \boldsymbol{B}\boldsymbol{z}^{(t)})$$

$$\boldsymbol{z}^{(t+1)} \leftarrow \arg\min_{\boldsymbol{z}\in\mathcal{Z}} g(\boldsymbol{z}) + \langle \boldsymbol{\lambda}^{(t)}, \boldsymbol{A}\boldsymbol{x}^{(t+1)} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{c}\rangle + \rho \mathrm{B}_{\phi}(\boldsymbol{B}\boldsymbol{z}, \boldsymbol{c} - \boldsymbol{A}\boldsymbol{x}^{(t+1)})$$

$$\boldsymbol{\lambda}^{(t+1)} \leftarrow \boldsymbol{\lambda}^{(t)} + \tau(\boldsymbol{A}\boldsymbol{x}^{(t+1)} + \boldsymbol{B}\boldsymbol{z}^{(t+1)} - \boldsymbol{c}),$$

where $\tau, \rho >$ are parameters of the solver. They also provide a convergence proof for BADMM under standard ADMM assumptions[2] and the condition that the applied Bregman divergence is α-strongly convex with respect to some p-norm, i.e., $\mathrm{B}_{\phi}(\boldsymbol{u}|\boldsymbol{v}) \geq \frac{\alpha}{2}\|\boldsymbol{u}-\boldsymbol{v}\|_p^2$.

2.4 Discrete optimal transport

Optimal transport (OT) has long been known by the mathematical community as a powerful tool for comparing probability measures (or distributions). Under different settings, it is also known as the *earth mover's distance* [37], or the *Wasserstein* distance. It was initiated by Monge [38] in 1781 and further developed by Kantorovich [39] in 1942 into what is known today as the Monge-Kantorovich formulation (or OT's primal formulation). In this section, we will present several "old and new" optimal transport formulations as well as introduce some of the computational and statistical aspects that will be used in subsequent chapters. We also refer the reader to excellent texts on the topic, namely Villani's quintessential monograph on a broad and comprehensive introduction to OT [40], Santambrogio's monograph on applied OT connections with PDEs [41], Luigi, Nicola and Giuseppe's "bible" on optimal transport gradient flows [42], and finally Cuturi and Peyré's recent but indispensable resource on computational aspects of optimal transport in modern settings [43].

We shall denote the optimal transport distance between vector arguments $\boldsymbol{\mu}, \boldsymbol{\nu}$ as $\mathcal{W}(\boldsymbol{\mu}, \boldsymbol{\nu})$. The signal arguments of interest shall be denoted as $\boldsymbol{\mu} \in \Sigma_m, \boldsymbol{\nu} \in \Sigma_n$, which we shall as-

[2]There exists a minimizer to the problem, and objective function is closed, proper and convex.

sume for now are probability measures lying in the probability simplex $\Sigma_n = \{\boldsymbol{p} \in \mathbb{R}_+^n : \mathbb{1}^\top \boldsymbol{p} = 1\}$. We will later describe more general OT formulations that can accept *unbalanced* arguments, i.e., $\mathbb{1}^\top \boldsymbol{\mu} \neq \mathbb{1}^\top \boldsymbol{\nu}$. The most widely used and general OT formulation, also known as Kantorovich's primal formulation or the *Monge-Kantorovich* formulation, is the simple constrained linear program given by:

$$\min_{\boldsymbol{P} \in \mathcal{U}(\boldsymbol{\mu}, \boldsymbol{\nu})} \langle \boldsymbol{P}, \boldsymbol{C} \rangle = \sum_{i=1}^{m} \sum_{j=1}^{n} P_{ij} C_{ij}. \tag{OT}$$

$\boldsymbol{P}$ is known as a *transport coupling* matrix (or correspondence matrix), whose i, j-th entry P_{ij} denotes the amount of "mass" that is to be transported between μ_i and ν_j. For mass conservation, we require $\boldsymbol{P}$ to be constrained in the set $\mathcal{U}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \{\boldsymbol{P} \in \mathbb{R}_+^{m \times n} : \boldsymbol{P}\mathbb{1} = \boldsymbol{\mu}, \boldsymbol{P}^\top \mathbb{1} = \boldsymbol{\nu}\}$ also known as the *transportation polytope*. This enforces $\boldsymbol{P}$ to be a doubly stochastic matrix whose marginals are the discrete distributions $\boldsymbol{\mu} \in \Sigma_m$ and $\boldsymbol{\nu} \in \Sigma_n$. The inner product between matrices are understood to be the standard Frobenius inner product, and $\boldsymbol{C} \in \mathbb{R}^{m \times n}$ is referred to as a *cost* metric that encodes geometry of the support. To this end, C_{ij} captures the distance between pairs of discrete locations along the support of the marginals. Intuitively, the program may be interpreted as finding the transportation configuration that minimizes the total work required to transport one histogram to another, under mass conservation constraints (i.e., mass cannot be destroyed nor created).

2.4.1 Eulerian and Lagrangian discretizations

We will present two important discrete representations for the input arguments of the OT program: the Eulerian and Lagrangian discretizations (representations).

In the Eulerian setting, $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are represented as *histograms* containing fixed discretization (bins) along the support's dimensions. This representation is typical in many fixed/regular grid measurement systems in signal processing (e.g., camera sensors, audio digitizers, radio-frequency digitizers). Here, the cost metric C_{ij} refers to the distance be-

tween the respective support locations μ_i and ν_j.

In the Lagrangian setting, $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are represented as *point clouds*. This representation is practical when dealing with data in high support-dimensions (e.g., multi-modal sensor systems) because the Eulerian representation rapidly becomes too massive for computation and storage. This representation is therefore typically used when working with empirical data distributions, where

$$\boldsymbol{\mu} = \sum_{i=1}^{m} p_i \delta_{\boldsymbol{x}_i}, \qquad \boldsymbol{\nu} = \sum_{j=1}^{n} q_j \delta_{\boldsymbol{y}_j}$$

each represent a weighted collection of data points. Each data point is denoted as a weighted point mass $\delta_{\boldsymbol{x}_i}, \delta_{\boldsymbol{y}_j}$ whose position given by data coordinates $\boldsymbol{x}_i, \boldsymbol{y}_j \in \mathbb{R}^d$ (e.g., latent embedding coordinates), along with importance weights given by p_i, q_j (with $\boldsymbol{p} \in \Sigma_m, \boldsymbol{q} \in \Sigma_n$).

2.4.2 Variants of optimal transport

Wasserstein distance

Suppose $\delta_i^{\mu}, \delta_j^{\nu}$ describe coordinates of the support of the signal w.r.t. $\boldsymbol{\mu}, \boldsymbol{\nu}$ respectively; i.e., δ_i^{μ} represents the i-th discrete support coordinate of the signal $\boldsymbol{\mu}$. When the cost metric $C_{ij} := d(\delta_i^{\mu}, \delta_j^{\nu})$ is generated with a valid metric $d(\cdot, \cdot)$ (i.e., satisfies all properties of a distance), the resulting OT formulation is also known as the p-Wasserstein distance

$$\mathcal{W}_p(\boldsymbol{\mu}, \boldsymbol{\nu}) \overset{\text{def.}}{=} \left(\min_{\boldsymbol{P} \in \mathcal{U}(\boldsymbol{\mu}, \boldsymbol{\nu})} \sum_{i=1}^{m} \sum_{j=1}^{n} P_{ij} d^p(\delta_i^{\mu}, \delta_j^{\nu}) \right)^{1/p}.$$

A popular choice for $d(\cdot, \cdot)$ is the Euclidean ℓ_2 distance (i.e., $d(\delta_i^{\mu}, \delta_j^{\nu}) \overset{\text{def.}}{=} \|\delta_i^{\mu} - \delta_j^{\nu}\|_2$) or the Euclidean ℓ_1 distance (i.e., $d(\delta_i^{\mu}, \delta_j^{\nu}) \overset{\text{def.}}{=} \|\delta_i^{\mu} - \delta_j^{\nu}\|_1$).

In image processing, the Wasserstein distance is a natural fidelity term in inverse problems when measurements are probability measures. Examples of Wasserstein applications

under the Eulerian representation are: image-restoration [44], tomography [45, 46], compressive sensing [47], matrix factorization [48, 49], image retrieval [37], while with the Lagrangian representation, some examples are with transfer learning and domain adaptation [50, 51], image color transfer [52].

Beckmann's formulation of the 1-Wasserstein distance

Optimal transport is notoriously expensive to compute since each evaluation is itself a linear optimization with $O(n^2)$ variables (i.e., size of $\boldsymbol{P}$). As recently shown by Li et al. [53], Beckmann's formulation provides an efficient reformulation of the optimal transport problem using an imcompressible fluid interpretation [54], dramatically reducing the number of optimization variables from $O(n^2)$ to $O(n)$, thereby extending its applicability to problems with high dimensionsal state spaces (e.g., imaging). A key assumption with Beckmann's reformulation is that it assumes Euclidean ground costs, i.e., $c(i,j) = \|\delta_i - \delta_j\|_2$ (or $\|\delta_i - \delta_j\|_1$), where δ_i represents the coordinates at support location i. Under these assumptions, mass cannot teleport and must follow straight-line paths between sources $\boldsymbol{\mu}$ and sinks $\boldsymbol{\nu}$. By modelling mass transport with a flux field, we dramatically reduce the number of optimization variables from $O(n^2)$ to $O(n)$. Beckmann's discrete optimal transport formulation may be stated as

$$\mathcal{W}_1(\boldsymbol{\mu}, \boldsymbol{\nu}) \equiv \min_{\boldsymbol{M}} \ \|\boldsymbol{M}\|_{2,1} \ \text{ s.t. } \ \text{div}(\boldsymbol{M}) - \boldsymbol{\mu} + \boldsymbol{\nu} = 0, \tag{2.5}$$

where $\boldsymbol{M} \in \mathbb{R}^{n \times D}$ denotes a (fluidic) flux field with D representing the dimensions of the field (i.e., support dimensions), and $\|\boldsymbol{M}\|_{2,1} = \sum_{i=1}^{n} \|\boldsymbol{M}_{i,:}\|_2$. In the case of images, $D = 2$, and columns of $\boldsymbol{M}$ may be reexpressed using $\boldsymbol{M}_x, \boldsymbol{M}_y \in \mathbb{R}^{n_x \times n_y}$ which represent the flux fields travelling in directions according to each dimension (with $n = n_x n_y$). The divergence of $\boldsymbol{M}$, notated as $\text{div}(\boldsymbol{M})$, measures how much a discrete point in the flux field is a source or a sink. According to context, we interchange its indexing notation

between column-major vector subscript $k \in \{1, \dots, n\}$, and pixel-wise coordinate indices $i \in \{1, \dots, n_x\}$ and $j \in \{1, \dots, n_y\}$ using external brackets:

$$\text{div}(\boldsymbol{M})_k \equiv \text{div}(\boldsymbol{M})[i,j] = (\boldsymbol{M}_x[i,j] - \boldsymbol{M}_x[i-1,j]) + (\boldsymbol{M}_y[i,j] - \boldsymbol{M}_y[i,j-1]),$$

with zero-flux boundary conditions (i.e., $\boldsymbol{M}_x[i,j] = \boldsymbol{M}_y[i,j] = 0$ if i or j lies outside the support). This formulation has recently been successfully applied in [53] for computing the OT cost between large scale images.

Partial and unbalanced optimal transport

Mass-preservation (balanced) constraints may be unrealistic in many practical applications when signals are not histograms or probability measures. For instance in videos, the physical preservation of mass is likely to hold for rigid objects in 3 spatial dimensions across time, but is certainly violated when the scene is viewed from some 2-d projection plane; mass preservation constraints at the projection plane is therefore a poor model of reality. Or consider a radar scenario where a target disappears from one frame to the next: normalization will arbitrarily and artificially increase energy of other targets to account for the target that has disappeared. Here we introduce two OT strategies, *partial* and *unbalanced* transport, that circumvent the modeling limitations of balanced constraints.

Partial OT [55, 37] limits its transportation budget to only a fraction of mass in its arguments, i.e., $\mathbb{1}^\top \boldsymbol{P} \mathbb{1} \leq \min(\mathbb{1}^\top \boldsymbol{\mu}, \mathbb{1}^\top \boldsymbol{\nu})$, and only transporting the minimum mass within the marginals (i.e., $\boldsymbol{P}\mathbb{1} \leq \boldsymbol{\mu}, \boldsymbol{P}^\top \mathbb{1} \leq \boldsymbol{\nu}$):

$$\min_{\boldsymbol{P} \geq 0} \ \langle \boldsymbol{P}, \boldsymbol{C} \rangle \ \text{ s.t } \ \boldsymbol{P}\mathbb{1} \leq \boldsymbol{\mu}, \ \ \boldsymbol{P}^\top \mathbb{1} \leq \boldsymbol{\nu}, \ \ \mathbb{1}^\top \boldsymbol{P} \mathbb{1} \leq \min(\mathbb{1}^\top \boldsymbol{\mu}, \mathbb{1}^\top \boldsymbol{\nu}). \tag{2.6}$$

Despite its concise and intuitive formulation, its lack of convexity (due to the last constraint) requires a reformulation for it to be suitable for variational settings (i.e., optimization is w.r.t. its arguments).

A recent strategy proposed by [56, 57] additionally models statistical properties of unaccounted mass via a mechanism of growth and decay. We define *unbalanced* transport as

$$\min_{\boldsymbol{P}\geq 0} \ \langle \boldsymbol{P}, \boldsymbol{C}\rangle + \mu(D(\boldsymbol{P}\mathbb{1}|\boldsymbol{\mu}) + D(\boldsymbol{P}^\top\mathbb{1}|\boldsymbol{\nu})), \tag{2.7}$$

where $D(\cdot|\cdot)$ denotes some notion of divergence, for which notable examples include:

- Squared ℓ_2-norm. $D(\boldsymbol{a}|\boldsymbol{b}) = \|\boldsymbol{a}-\boldsymbol{b}\|_2^2$ (first introduced in [58]). This models mass growth/decay as a Gaussian process.
- ℓ_1-norm (or total variation distance). $D(\boldsymbol{a}|\boldsymbol{b}) = \|\boldsymbol{a}-\boldsymbol{b}\|_1$, which is equivalent to the Lagrangian formulation of partial OT [55, 59] (see [56, 60] for more details). This models mass growth/decay as a Laplacian process.
- Kullbeck-Liebler divergence. $D(\boldsymbol{a}|\boldsymbol{b}) = \mathrm{KL}(\boldsymbol{a}|\boldsymbol{b})$, proposed by [57, 61], which is closely related to other statistical divergences (see [43, § 10.2]).

The terms weighed by parameter $\mu > 0$ penalize unaccounted mass between the marginals of the optimal transport coupling and the input arguments, which regulates growth/decay.

2.4.3 Computational methods

Traditional approaches

The (OT) problem is a linearly constrained linear program (n^2 variables and $2n$ constraints, where n denotes size of its input arguments) which can be solved with methods such as interior point methods (IPMs) or the simplex method, which have a computational complexity of $O(n^3 \log n)$ [43] per iteration. Since the solution of (OT) lies on the convex hull of the transportation polytope, interior point methods erect a log-barrier to successively traverse the interior of transportation polytope, while the simplex method directly traverses along the polytope. Compared to the simplex method which is efficient at evaluating (OT) only in isolation, off-the-shelf IPMs (e.g., MOSEK, or Gurobi [62]) facilitate variational

OT formulations (OT may be used as a loss where variability is in the OT's arguments). IPMs have very efficient convergence rates (often converging within tens of iterations regardless of problem size), but incur high per-iteration costs due to a dense $2n \times 2n$ matrix inversion per iteration [20]. As a result, they scale poorly for huge problem sizes, necessitating first order methods that have poorer convergence rates but have much more tractable per-iteration costs.

Entropic regularization

Cuturi, in his landmark paper [63], demonstrated a fast and accurate approximation to the Monge-Kantorovich OT formulation via (negative) entropic regularization. The resultant *Sinkhorn-Knopp* algorithm [64] (i.e., iterative proportional fitting procedure) is extremely simple to implement and is embarrassingly parallelizable. It was later shown [65] that this algorithm was closely related to the more general concept of Bregman (KL) iterations [33], which is a precursor to efficient algorithms like mirror descent [34, 35]. While a detailed derivations can be found in [63, 65, 43], we present a short derivation to make explicit a few important connections between Bregman iterations and entropic regularization.

Writing (OT) with negative entropic regularization yields the following problem:

$$\min_{\boldsymbol{P} \in \mathcal{U}\boldsymbol{\mu},\boldsymbol{\nu}} \langle \boldsymbol{P}, \boldsymbol{C} \rangle + \varepsilon \langle \boldsymbol{P}, \log \boldsymbol{P} \rangle , \tag{2.8}$$

where $\varepsilon > 0$ is the regularization parameter, and $\log \boldsymbol{P}$ is taken elementwise. The greater the negative entropic regularization (i.e., larger ε), the more the doubly stochastic matrix $\boldsymbol{P}$ tends towards a uniform distribution; conversely, as $\varepsilon \to 0$, (2.8) tends towards the original problem (OT). The introduction of the negative entropy can also be viewed as a smoothing term that strongly convexifies the optimization.

Manipulating its Lagrangian reveals that the algebraic structure of an entropic barrier over the transportation polytope admits an alternating Bregman (KL) projection algorithm.

Writing its Lagrangian:

$$\mathcal{L}(\boldsymbol{P}, \boldsymbol{a}, \boldsymbol{b}) = \langle \boldsymbol{P}, \boldsymbol{C} \rangle + \varepsilon \langle \boldsymbol{P}, \log \boldsymbol{P} \rangle + \langle \boldsymbol{a}, \boldsymbol{P}\mathbb{1} - \boldsymbol{\mu} \rangle + \langle \boldsymbol{b}, \boldsymbol{P}\mathbb{1} - \boldsymbol{\nu} \rangle ,$$

where $\boldsymbol{a}, \boldsymbol{b}$ are Lagrange multipliers. The first order stationarity conditions w.r.t. $\boldsymbol{P}$ yields

$$\boldsymbol{C} + \varepsilon \log \boldsymbol{P} + \boldsymbol{a}\mathbb{1}^\top + \mathbb{1}\boldsymbol{b}^\top = 0 \;\; \Rightarrow \;\; \boldsymbol{P} = \exp(-\boldsymbol{a}\mathbb{1}^\top/\varepsilon)\boldsymbol{K}\exp(-\mathbb{1}\boldsymbol{b}^\top/\varepsilon),$$

where $\boldsymbol{K} \overset{\text{def.}}{=} \exp(-\boldsymbol{C}/\varepsilon)$. According to Sinkhorn's theorem [64], since $\boldsymbol{K}$ is strictly positive, there exists a *unique* matrix $\boldsymbol{P}$ of the form $\exp(\boldsymbol{u})\boldsymbol{K}\exp(\boldsymbol{v})$ that belongs to $\mathcal{U}(\boldsymbol{\mu}, \boldsymbol{\nu})$, where $\boldsymbol{u}, \boldsymbol{v} \geq 0$. $\boldsymbol{P}$ is solved with a surprisingly simple iterative method called the Sinkhorn-Knopp algorithm or the iterative proportional fitting procedure (IPFP), which we summarize in Algorithm 1.

Algorithm 1 Sinkhorn-Knopp algorithm [64]

procedure SINKHORN($\gamma > 0, \boldsymbol{\mu} \in \Sigma_m, \boldsymbol{\nu} \in \Sigma_n, \boldsymbol{C} \in \mathbb{R}_+^{m \times n}$)
  $\boldsymbol{K} \leftarrow \exp(-\boldsymbol{C}/\gamma), \quad \boldsymbol{v} \leftarrow \frac{\mathbb{1}_n}{n}$
  while not converged **do**
    $\boldsymbol{u} \leftarrow \boldsymbol{\mu} \oslash \boldsymbol{K}\boldsymbol{v}$
    $\boldsymbol{v} \leftarrow \boldsymbol{\nu} \oslash \boldsymbol{K}^\top \boldsymbol{u}$
  end while
  $\boldsymbol{P} \leftarrow \operatorname{diag}(\boldsymbol{u})\boldsymbol{K}\operatorname{diag}(\boldsymbol{v})$
end procedure

In subsequent work by Benamou et al. [65], it was revealed that an entropically regularized linear program under transportation constraints could be solved efficiently with iterative Bregman projections, for which the Sinkhorn algorithm was a special case of. The main insight was that the entropically regularized objective of (2.8) could be reframed as a KL-divergence between $\boldsymbol{P}$ and a Gibbs kernel (i.e., $\boldsymbol{K}$), and constraints could be cast as iterative projections. Defining the regularization term slightly differently as

$E(\boldsymbol{P}) \overset{\text{def.}}{=} \left\langle \boldsymbol{P}, \log \boldsymbol{P} + \mathbb{1}\mathbb{1}^\top \right\rangle + \iota_+(\boldsymbol{P})$, the objective may be reexpressed as

$$\min_{\boldsymbol{P} \in \mathcal{U}(\boldsymbol{\mu}, \boldsymbol{\nu})} \langle \boldsymbol{P}, \boldsymbol{C} \rangle + \varepsilon E(\boldsymbol{P}) = \min_{\boldsymbol{P} \in \mathcal{U}(\boldsymbol{\mu}, \boldsymbol{\nu})} \mathrm{KL}(\boldsymbol{P} | \boldsymbol{K}),$$

where $\boldsymbol{K} \overset{\text{def.}}{=} \exp(-\boldsymbol{C}/\varepsilon)$. [65] showed that since constraints $\mathcal{U}(\boldsymbol{\mu}, \boldsymbol{\nu})$ were in fact a union of convex constraints, i.e. $\mathcal{U} = \bigcup_{i=1,2} \ \mathcal{U}_i$ such that

$$\mathcal{U}_1 \overset{\text{def.}}{=} \{\boldsymbol{P} \in \mathbb{R}_+^{m \times n} : \boldsymbol{P}\mathbb{1} = \boldsymbol{\mu}\}, \qquad \mathcal{U}_2 \overset{\text{def.}}{=} \{\boldsymbol{P} \in \mathbb{R}_+^{m \times n} : \boldsymbol{P}^\top \mathbb{1} = \boldsymbol{\nu}\},$$

it admits the following convergent (and concise) sequence of iterative KL projections

$$\boldsymbol{P}^{(t)} \leftarrow P_{\mathcal{U}_t}^{\mathrm{KL}}(\boldsymbol{P}^{(t-1)}) \overset{\text{def.}}{=} \operatorname*{arg\,min}_{\boldsymbol{P} \in \mathcal{U}_t} \ \mathrm{KL}(\boldsymbol{P} | \boldsymbol{K}), \qquad \text{where} \qquad \mathcal{U}_{t+2} = \mathcal{U}_t,$$

whose projections on sets $\mathcal{U}_1, \mathcal{U}_2$ may be derived using exponentiated gradients as

$$P_{\mathcal{U}_1}^{\mathrm{KL}}(\boldsymbol{P}) = \operatorname{diag}\left(\frac{\boldsymbol{\mu}}{\boldsymbol{P}\mathbb{1}}\right)\boldsymbol{P}, \qquad P_{\mathcal{U}_2}^{\mathrm{KL}}(\boldsymbol{P}) = \boldsymbol{P} \operatorname{diag}\left(\frac{\boldsymbol{\nu}}{\boldsymbol{P}^\top \mathbb{1}}\right). \tag{2.9}$$

While this framework is very versatile (it allows multiple types of constraints to be easily incorporated), it has several limitations, namely

- it approximates (OT) rather than solving it precisely (due to the presence of a regularization term),
- the decomposition format (of KL projections) limits its application to a narrow class of objective functions.

Bregman ADMM

Using Wang and Banerjee's Bregman ADMM approach [36], problem (OT) may be cast as

$$\min_{\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{R}_+^{m \times n}} \langle \boldsymbol{P}, \boldsymbol{C} \rangle + \iota_{\mathcal{U}_1}(\boldsymbol{P}) + \iota_{\mathcal{U}_2}(\boldsymbol{Q}) \qquad \text{s.t.} \qquad \boldsymbol{P} = \boldsymbol{Q},$$

where the constraint sets $\mathcal{U}_1, \mathcal{U}_2$, respectively, are

$$\mathcal{U}_1 \overset{\text{def.}}{=} \{\boldsymbol{P} \in \mathbb{R}_+^{m \times n} : \boldsymbol{P}\mathbb{1} = \boldsymbol{\mu}\}, \qquad \mathcal{U}_2 \overset{\text{def.}}{=} \{\boldsymbol{P} \in \mathbb{R}_+^{m \times n} : \boldsymbol{P}^\top \mathbb{1} = \boldsymbol{\nu}\}.$$

The Bregman ADMM algorithm admits the following update steps under a KL divergence:

$$\begin{aligned}
\boldsymbol{P}^{(t+1)} &\leftarrow \operatorname*{arg\,min}_{\boldsymbol{P} \in \mathcal{U}_1} \ \langle \boldsymbol{P}, \boldsymbol{C} \rangle + \langle \boldsymbol{A}^{(t)}, \boldsymbol{P} \rangle + \rho \mathrm{KL}(\boldsymbol{P} | \boldsymbol{Q}^{(t)}) = \exp(-(\boldsymbol{C} + \boldsymbol{A}^{(t)})/\rho + \log \boldsymbol{Q}^{(t)}) \\
\boldsymbol{Q}^{(t+1)} &\leftarrow \operatorname*{arg\,min}_{\boldsymbol{Q} \in \mathcal{U}_2} \ \langle \boldsymbol{A}^{(t)}, -\boldsymbol{Q} \rangle + \rho \mathrm{KL}(\boldsymbol{Q} | \boldsymbol{P}^{(t+1)}) = \exp(-\boldsymbol{A}^{(t)}/\rho + \log \boldsymbol{P}^{(t+1)}) \\
\boldsymbol{A}^{(t+1)} &\leftarrow \boldsymbol{A}^{(t)} + \tau(\boldsymbol{P}^{(t+1)} - \boldsymbol{Q}^{(t+1)}).
\end{aligned}$$

We note that these update steps have closed form solutions derived as exponentiated gradients, which are similar to the KL projections (2.9) of [65]. The key difference between the two is that with BADMM, the updates are Bregman proximal steps instead of Bregman projections; this means that they have the flexibility to incorporate more than just the indicator functions (for set constraints). One feature/limitation of KL proximal steps is this: their geometry produces gradient steps that are specifically optimized for signals in the simplex [34, 35], hence generic signals (not in the simplex) could suffer from arbitrarily slow convergence rates.

2.4.4 Statistical properties

Given empirical measures, $\widehat{\boldsymbol{\alpha}} = \frac{1}{n}\sum_{i=1}^n \delta_{\boldsymbol{x}_i}$ and $\widehat{\boldsymbol{\beta}} = \frac{1}{m}\sum_{j=1}^m \delta_{\boldsymbol{y}_j}$, where $\delta_{\boldsymbol{x}}$ refers to a point mass located at coordinates given by $\boldsymbol{x}$, we say the statistical divergence D is a *consistent* estimator if, as $m, n \to +\infty$,

$$D(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}) \to D(\boldsymbol{\alpha}, \boldsymbol{\beta}).$$

It was shown by Dudley [66] that for measures supported on a bounded domain in $\mathbb{R}^d$, the p-Wasserstein distance $\mathcal{W}_p$ was a consistent estimator with a rate of convergence (i.e.,

sample complexity) given by

$$\mathbb{E}(|\mathcal{W}_p(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}) - W_p(\boldsymbol{\alpha}, \boldsymbol{\beta})|) = O(n^{-1/d}),$$

for $d > 2$ and $1 \leq p \leq \infty$. Although the p-Wasserstein distance is a non-parametric estimator (i.e., it doesn't require estimation of the kernel) that exploits geometry of the support, it suffers from the curse of dimensionality because its sample complexity grows exponentially with dimensions d. More recently, work by Weed and Bach [67] showed that when the measures are approximately supported on low-dimensional subdomains, the rate of convergence depends on the intrinsic dimension rather than the ambient dimension.

When the Wasserstein distance is entropically-regularized, it is known as the Sinkhorn distance (2.8) which interpolates between the *maximum mean discrepancy* (MMD)[3] [68] and the p-Wasserstein distance. As shown by Genevay et al. [69], when $p = 2$, the Sinkhorn distance's sample complexity interpolates between the two rates

[3]The MMD is a consistent divergence measure whose sample complexity does not depend on the ambient dimension, i.e., $O(n^{-1/2})$.

CHAPTER 3

A REAL-TIME APPLICATION IN MICROSCOPY DECONVOLUTION

Live cell imaging allows the monitoring of complex biophysical phenomena as they happen in real time, which is beneficial in studying biological functions, observing drug action, or monitoring disease progression. For these experiments, tissue from an organ such as brain, heart, or liver is sliced and imaged while it is still alive [70, 71, 72, 73]. Fluorescence microscopy is often used for live cell imaging but is not always practical because it requires the use of dyes or genetic engineering techniques to introduce fluorophores into the sample. Instead, it is often desirable to image unlabeled, otherwise optically transparent samples. This is often done using a phase contrast-enhancing technique such as differential interference contrast (DIC) microscopy.

Using machine vision to automatically segment individual cells under DIC optics in real time would be highly useful for microscopy automation of life science experiments. However, precise cell segmentation is challenging and the vast majority of existing algorithms are not directly applicable to segmentation under DIC in tissue. General purpose segmentation algorithms in the computer vision literature typically assume statistical homogeneity within (or outside) a segmentation region that would be lost under contrast-enhancing optical approaches such as DIC. While it is possible that these algorithms could be applied after application-specific pre-processing [74, 75], these existing approaches are not directly applicable to the target application without being combined with deconvolution. There are additionally a number of specific cell segmentation and tracking methods that [76, 77, 78, 79, 80, 81] are also not directly applicable to cell segmentation under DIC in tissue. For example, algorithms proposed in the Cell Tracking Challenge [82, 83] are developed on the CTC dataset that consists only of microscopy imagery of cultured cells (rather than tissue slices) that had minimal organic tissue interference (Fig. 3.1). These cell-tracking

methods (i) often assume simple noise statistics, (ii) target gross cell location tracking for mechanobiology tasks (e.g., studies on cell migration, morphology) rather than precise membrane localization, and (iii) are designed to be run offline rather than in real-time [84].

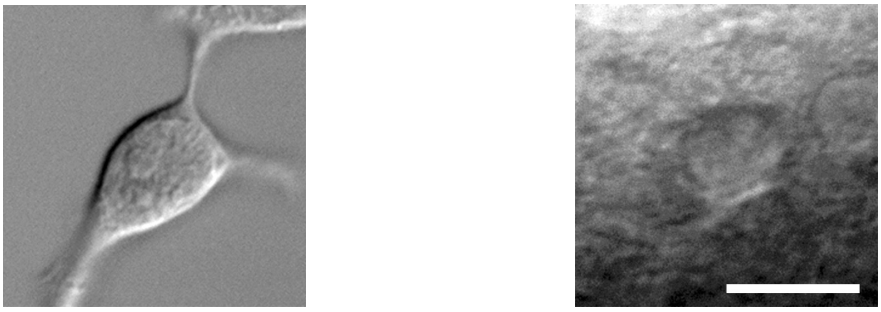


Figure 3.1: Images obtained with DIC microscopy (40x magnification). Left: cultured human embryonic kidney (HEK293) cell. Right: neuron in mouse brain tissue (400 um thickness). The image on the right has high levels of imaging noise due to light scatter and interference due to organic material in surrounding tissue. Scale bar 10 μm.

We present the first cell segmentation and boundary tracking algorithm for DIC imagery of tissue slices. While our broad approach may generalize to other tasks, we focus here on one experiment type: patch-clamp recording in brain tissue. In this experimental paradigm, brain tissue is sliced into $100 - 400\mu m$ thick sections, each containing thousands of neurons embedded in a dense biological milieu. The slice is imaged with a microscope and glass probe is inserted into the tissue to sample the electrical activity of individual cells [85, 86]. While manual patch clamp electrophysiology is considered to be the gold-standard for high-fidelity single-cell analysis, the challenge and labor intensity of the process makes it extremely advantageous to automate. Recent work has demonstrated the possibility of automating the patch clamp process by using a motorized robotic actuator to maneuver the probe to the target cell [87, 88]; for this purpose, real-time tracking of the target cell boundary is essential. This application presents several challenges that make cell membrane localization very difficult: (i) heavy interference from the presence of organic tissue around the target cell, (ii) low SNR due to scattering of light characteristic of thick tissue samples, and (iii) cell motion induced by the glass probe.

In this chapter, we present four main contributions[1]: (i) a realistic cell simulator for DIC microscopy imagery, (ii) a Bayesian inference approach to DIC deconvolution for cell membrane tracking in tissue, (iii) a solver for real-time system deployment using parallel hardware (GPU), and (iv) software integration with a robotic patch-clamp experimental rig.

Realistic cell simulator. We describe new community infrastructure in the form of a MATLAB toolbox for accurately simulating DIC microscopy images of *in vitro* brain slices. Building on existing DIC optics modeling, our simulation framework additionally contributes an accurate representation of interference from organic tissue, neuronal cell-shapes, and tissue motion due to the action of the pipette. This simulator allows us to better understand the image statistics to improve algorithms, as well as quantitatively test cell segmentation and tracking algorithms in scenarios where ground truth data is fully known.

Membrane tracking algorithm. Extending the deconvolution framework provided by Li and Kanade [96], our proposed algorithm is formulated as a regularized least-squares optimization that incorporates a filtering mechanism to handle organic tissue interference and a robust edge-sparsity regularizer that integrates dynamic edge-tracking capabilities. Methodologically inspired by Charles et al. [97], we develop an iterative *reweighted* algorithm derived using a hierarchical Bayesian probabilistic model. We specifically note that the proposed algorithm is performing a deconvolution of the complex effects of DIC imaging integrated into a segmentation process and is not a direct segmentation of raw DIC images. Toward this end, our focus is specifically on cell boundary tracking instead of more typical deconvolution metrics such as least-square image reconstruction.

Real-time GPU implementation. While the previous section tests the efficacy of the

[1]This work was performed in collaboration with Christopher J. Rozell, Ilya Kolb, Craig Forest, Corey R Landry, Mighten C Yip, Colby F Lewallen, William A Stoy, Amanda Felouzis, Bo Yang, and Edward S Boyden. With regards to the algorithm development, IK was responsible for data collection, while JL developed the cell simulator framework, formulated the membrane tracking algorithm [89, 90, 91], and designed/implemented the solver in GPU hardware [92]. JL and CJR are inventors on a US patent application 16/116,192 related to cell membrane tracking in tissue [93]. With regards to work related to the *PatcherBot* system [94, 95], IK, WAS, ESB and CRF conceived the project and experimental design. IK built the PatcherBot system with contributions by MCY and CFL. IK, CFL and MCY performed the validation experiments. JL and CJR developed the cell tracker. AF performed pilot Tergazyme experiments. BY prepared brain slices and assisted in experiments. ESB, CJR, and CRF supervised the project.

proposed membrane tracking algorithm, we require it to be practical. Namely, we need a fast algorithm to meet the real-time requirements of the robotic system. Reweighted sparsity approaches are known to be effective and robust for signal reconstruction, yet costly because they require solving an optimization per reweight. We build on the alternating directions method of multipliers (ADMM), which is a scalable approach for such image inverse problems. We expand on existing ADMM frameworks for deconvolution and inpainting to develop a fast RWTV-ADMM solver framework which has near-linear time complexity of $O(n \log n)$ operations per iteration (with n representing the image size). In particular, the iterative nature of both the ADMM algorithm and RWTV was exploited to develop an efficient method that integrates the iterations. In addition, we develop a GPU-accelerated implementation of the proposed solver and demonstrate its efficiency for a real-time application in microscopy cell deconvolution and segmentation.

Integration in automated patch-clamp system. The development of automated patch clamping is transformative for applications that depend on many high-quality measurements of single cells, such as drug screening, protein functional characterization, and multimodal cell type investigations. We integrate our membrane tracking algorithm into a robotic patch clamping rig developed by Dr. Forest's lab [87] with the main goal to improve experimental patch clamp yield in *in-vitro* brain slices with computer vision. To this end, feedback from a sensing module (i.e., computer vision) is critical since it corrects for displacement errors due to an accumulation of actuator errors and cell motion.

3.1 Background and related work

3.1.1 DIC microscopy

DIC microscopy enhances the contrast of an image by exploiting the fact that differences in the tissue will have different optical transmission properties that can be measured through the principle of interferometry. Specifically, the signal that we aim to reconstruct is known as the optical path length (OPL) signal image, which is proportional to the the underlying

phase-shift. The OPL is defined as the product of refractive index and thickness distributions of the object relative to the surrounding medium [96, 98]. This OPL signal shall be denoted as $\boldsymbol{x}_k \in \mathbb{R}^N$ (a vectorized form of the $\sqrt{N} \times \sqrt{N}$ OPL image's intensity values), and indexed in time by subscript k. DIC microscopy amplifies differences between the cell's refractive index and its environment to enhance visibility of the cell, highlighting edge differentials and giving the appearance of 3D relief. This effect may be idealized as a convolution between the optics' point spread function and the OPL, and denoted as $\boldsymbol{D}\boldsymbol{x}_k$, where $\boldsymbol{D} \in \mathbb{R}^{N \times N}$ is a matrix that captures the 2D convolution against a kernel $\boldsymbol{d} \in \mathbb{R}^{K \times K}$.

While more sophisticated DIC imaging models (c.f., [99]) exist, for simplicity we use a kernel $\boldsymbol{d}$ corresponding to an idealized DIC model proposed by Li and Kanade [96] which is a steerable first-derivative of Gaussian kernel:

$$d(x, y) \propto -x \cos(\theta) \cdot e^{-\frac{x^2+y^2}{\sigma_d^2}} - y \sin(\theta) \cdot e^{-\frac{x^2+y^2}{\sigma_d^2}}, \tag{3.1}$$

where σ_d refers to the Gaussian spread and θ_d refers to a steerable shear angle. This model assumes an idealized effective point spread function (EPSF), where the condenser lens is infinite-sized and the objective is infinitely small (with respect to the wavelength). This model also ignores any phase wrapping phenomenon by assuming that the specimen is thin enough or the OPL varies slowly enough such that it behaves in the linear region of the phase difference function [99]. In practice, $d(x, y)$ is discretized as $\boldsymbol{d}$ with a proportionality constant of 1, and with x and y each taking the discrete domain $\{-\frac{K-1}{2}, -\frac{K-3}{2}, ..., \frac{K-3}{2}, \frac{K-1}{2}\}$.

3.1.2 Deconvolution Algorithms

DIC cell segmentation algorithms fall broadly into three categories: direct, machine-learned, or deconvolution (or phase-reconstruction) algorithms. Direct algorithms apply standard image processing operations such as low-pass filtering, thresholding, and morphological

shape operations [74, 75, 100] but are not robust and work only on very low-noise/interference imagery. Machine-learned algorithms perform statistical inference learned from a large number of cell-specific training images (e.g., deep convolutional networks [101] or Bayesian classifiers [102]). Such algorithms have been shown to perform coarse segmentation surprisingly well in challenging scenarios (e.g., cells with complicated internal structures with low-noise/interference) for applications like cell-lineage tracking or cell-counting, yet they appear to lack precision for accurate cell-boundary localization. Deconvolution algorithms [103, 104, 99, 105, 106] may be defined as a reconstructive process which estimates the OPL (or phase-shift image) that has been convolved by the DIC microscope's optical point spread function. Early work involved closed-form reconstructive methods such as deconvolution by Wiener filtering [99] or the Hilbert transform [106, 104] which were not robust against noise. Other work involved DIC deconvolution by constraining the support of deconvolution along a single line along the shear axis [105]. While computationally quick, this made the reconstruction prone to discontinuities between lines (especially in high-noise/interference images), resulting in significant streaking artifacts which corrupt the cell boundary estimates. Deconvolution (in 2D) was proposed in [103] by assuming some linear approximation to the point spread function optical model, yet this formulation lacked regularization, thus making the problem ill-posed and also highly susceptible to noise.

Recent developments in fast and robust ℓ_1 (i.e., $\|\cdot\|_1$) reconstruction approximations motivated Li and Kanade to develop a mixed-norm pre-conditioning approach [96] that exploited sparsity and smoothness in the OPL image using ℓ_1-norm and total variation (TV) norm (i.e., $\|\cdot\|_{\text{TV}}$) regularizers:

$$\widehat{\boldsymbol{x}}_k = \underset{\boldsymbol{x}\geq\boldsymbol{0}}{\arg\min}\ \frac{1}{2}\left\|\boldsymbol{y}_k - \boldsymbol{D}\boldsymbol{x}\right\|_2^2 + \beta\left\|\boldsymbol{x}\right\|_1 + \gamma\left\|\boldsymbol{x}\right\|_{\text{TV}}, \tag{3.2}$$

where β, γ are sparsity and smoothness parameters that control the weight of pixel sparsity and edge-sparsity against reconstruction fidelity respectively. The same paper proposed an

alternative approach that replaced the TV norm by a Laplacian Tikhonov term to make it more computationally efficient, but placed less emphasis on smoothness:

$$\widehat{\boldsymbol{x}}_k = \underset{\boldsymbol{x}\geq\mathbf{0}}{\arg\min}\ \frac{1}{2}\left\|\boldsymbol{y}_k - \boldsymbol{D}\boldsymbol{x}\right\|_2^2 + \beta\left\|\boldsymbol{\Lambda}\boldsymbol{x}\right\|_1 + \gamma\left\|\boldsymbol{L}\boldsymbol{x}\right\|_2^2, \tag{3.3}$$

where $\boldsymbol{\Lambda}$ is a positive weighting diagonal matrix (which may be optionally applied to further encourage pixel-sparsity), and $\boldsymbol{L}$ is a matrix that applies a 2D convolution of a Laplacian filter against the image.

A recent deconvolution algorithm intended to facilitate cell segmentation [107] advanced the work in [96] by reducing the computational complexity as well as introducing a dynamical prior (exploiting temporal structure) and a re-weighting process that improves reconstruction accuracy. The core algorithm can be summarized by two steps, an optimization step and a reweighting step as follows:

$$\begin{gathered}\widehat{\boldsymbol{x}}_k^{(t)} = \underset{\boldsymbol{x}\geq\mathbf{0}}{\arg\min}\ \frac{1}{2}\left\|\boldsymbol{y}_k - \boldsymbol{D}\boldsymbol{x}\right\|_2^2 + \beta\left\|\boldsymbol{\Lambda}^{(t-1)}\boldsymbol{x}\right\|_1 + \gamma\boldsymbol{x}^\top\boldsymbol{J}\boldsymbol{x} + \kappa(\boldsymbol{x}-\widehat{\boldsymbol{x}}_{k-1})^\top\boldsymbol{\Sigma}(\boldsymbol{x}-\widehat{\boldsymbol{x}}_{k-1}), \\ \boldsymbol{\Lambda}^{(t)}[i,i] = \frac{1}{\widehat{\boldsymbol{x}}_k^{(t)}[i]+\eta},\end{gathered} \tag{3.4}$$

where $\boldsymbol{\Lambda}$ is a positive diagonal matrix defining the ℓ_1 weights, $\boldsymbol{J}$ is a Laplacian matrix defining similarity between spatial neighbors, $\boldsymbol{\Sigma}$ is a matrix defining the similarity between temporal pixel neighbors, t represents the algorithmic iteration index, κ is a parameter that controls the influence of the dynamical regularizer, and η is a small positive constant that prevents division by zero. These methods were effective at retrieving the OPL for noisy microscopy images with little interference. However, organic tissue interference surrounding the cell negatively affected the reconstruction (and subsequently segmentation), especially around the edges of the cell (later demonstrated in Sections 3.3.4 and 3.3.6).

3.1.3 Cell Simulator

Accurate cell simulators are a valuable tool for two reasons: they allow objective testing with known ground truth, and provide insights into generative models that facilitate algorithm development. Currently, the vast majority of existing cell simulator packages focus specifically on fluorescence microscopy [108, 109, 110, 111, 112] rather than DIC microscopy. While some simulators [110, 109] excelled at providing a large variety of tools for simulating various experimental scenarios and setups, most lacked simulation of synthetic cellular noise similar to that found in DIC microscopy images of brain slices (due to the presence of cellular tissue). Most simulators tend to target very specific types of cells [108, 109, 112] and there have been initial efforts to organize and share cellular information (e.g., spatial, shape distributions) into standardized formats across simulators [15], [16]. Despite this, no existing simulator currently generates synthetic DIC imaging of neurons such as those used in patch clamp experiments for brain slices. To facilitate algorithm design and evaluation on the important problem of automated patch clamping, we have built and released MATLAB toolbox for accurately simulating DIC microscopy images of *in vitro* brain slices.

3.2 Realistic cell simulator

Most cell simulators can be described as having three distinct stages: cell-shape generation, optical imaging, and noise generation. The simulation framework implemented for this work uses these same three stages (shown in Fig. 3.2), adapting general approaches used in the literature for each stage so that the simulated data reflects the statistics of the DIC microscopy images for *in vitro* brain slice electrophysiology. In the first stage, a synthetic cell-shape is generated and embedded into the pixel-space as an ideal OPL image (i.e., the ground truth). In the second stage, the OPL image is transformed using a convolution against an idealized point spread function that approximately describes the DIC

microscope's optics. Lastly, interference from static components (e.g., organic material) and dynamic noise components (e.g., the image acquisition system) are generated and incorporated into the image during the third stage. In this work we have used existing general approaches to build and validate the model components in the simulator using real DIC imagery of in vitro brain slices from adult (P50-P180) mice as described in [113]. Note that this simulator is unique in that it is tailored to simulate this type of cells.

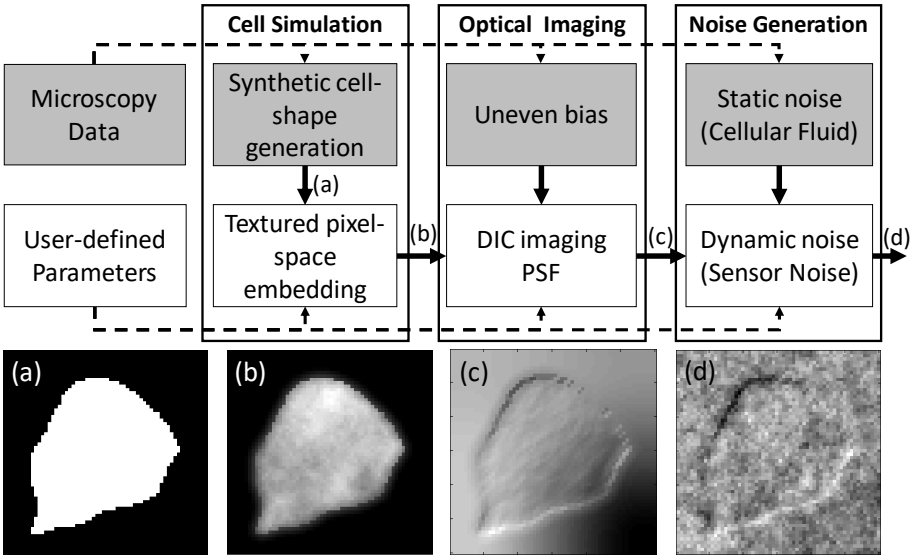


Figure 3.2: Block diagram of the cell simulator. Gray blocks: generative models that learn from available data. White blocks: generative models that rely on user-parameters. The respective images are outputs from the various stages of the simulator, including (a) Binary image of synthetic cell shape (b) Textured OPL image showing light transmission through tissue (c) Received image through DIC optics (d) Final image with noise and interference.

3.2.1 Cell shape generation

The cell-shape is unique to different applications and plays a significant role in algorithm development, necessitating customization in cell simulations [108, 114]. Extensive work on generic shape representation in [114] demonstrated that applying principal components analysis (PCA) on cell-shape outlines is an excellent strategy for reconstructing cell shapes. In this work, we applied a similar approach as [108] to generate synthetic cell-shapes using

PCA and multivariate kernel density estimation sampling on subsampled cell contours. For shape examples, we used hand drawn masks of neurons from DIC microscopy images of rodent brain slices.

Example cell-shapes are collected such that K coordinates are obtained in clockwise continuous fashion (around the contour), beginning at the north-most point. These points are centered such that the centroid is at the origin. The (x, y) coordinates are concatenated into an $\mathbb{R}^d$ vector

$$\boldsymbol{x}_i = (x_1, \ldots, x_K, y_1, \ldots, y_K)^\top,$$

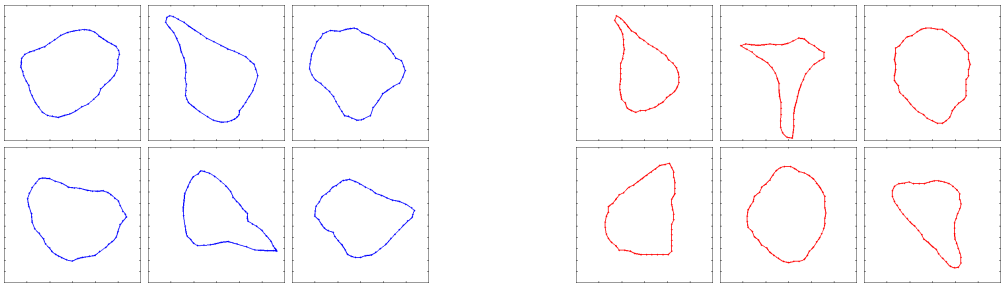
where $d = 2K$. This vector is then normalized via $\widehat{\boldsymbol{x}}_i = \boldsymbol{x}_i / \|\boldsymbol{x}_i\|_2$. All N normalized examples are gathered into the following matrix

$$\boldsymbol{X} = (\widehat{\boldsymbol{x}}_1, \ldots, \widehat{\boldsymbol{x}}_N).$$

Eigen-decomposition is performed on the data covariance matrix formed as $\boldsymbol{S} = \boldsymbol{X}\boldsymbol{X}^T = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$. Cell-shapes may thus be expressed with the coefficient vector $\boldsymbol{b}_i$ and the relationship given by

$$\boldsymbol{x}_i = \boldsymbol{V}\boldsymbol{b}_i \Leftrightarrow \boldsymbol{b}_i = \boldsymbol{V}^\top \boldsymbol{x}_i. \tag{3.5}$$

Since PCA guarantees that $\mathrm{cov}(b_i, b_j) = 0, \forall i \neq j$, kernel density estimation is performed individually on each of the coefficients to estimate its underlying distributions. This allows us to randomly sample from these distributions to produce a synthetically generated coefficient vector, $\widetilde{\boldsymbol{b}}$. The cell-shape may then be trivially converted into coordinates using the relationship given in (3.5). A rotation (θ_{rot}) and scaling (γ_{scale}) are added to the cell-shapes where necessary. Fig. 3.3 shows a sampling of synthetic and actual cell-shapes extracted from the data.



(a) (b)

Figure 3.3: Visual similarities were observed between the (a) synthetically generated cell-shapes, and the (b) cell-shapes extracted from DIC imagery of rodent brain slices.

Textured OPL pixel-space embedding

The previously generated cell-shape is embedded into the pixel space $f_{x,y}$ by a texture generation method similar to methods found in other fluorescence microscopy simulators [115, 116]. In this stage, the cell-shape is embedded into the pixel space with the generation of a textured OPL image. Perlin noise [117] is a well-established method for generating synthetic cell textures in fluorescent microscopy [115, 116], and we apply a similar concept because it generates realistic looking cell textures.

First, a binary mask, $m(x, y)$, is generated using the cell-shape's coordinates as polygon vertices and cast into an $M \times M$ image. Next, a textured image, $t(x, y)$ is generated by frequency synthesis; a 2-D filter with a $1/f^p$ frequency spectra is applied to white Gaussian noise, where p is a persistence term which controls the texture's heterogeneity. $t(x, y)$ is re-scaled such that $\widehat{t}(x, y) \in [0, 1]$. Finally, the OPL image is constructed by

$$f(x, y) = \big(m(x, y) * h(x, y)\big) \cdot \big(m(x, y) \cdot \widehat{t}(x, y)\big),$$

where $*$ signifies 2D convolution, and $h(x, y)$ is a 2-D filter (e.g., a circular pillbox averaging filter) that rounds the edges of the OPL image. An example surface texture is simulated and shown in Fig. 3.2(b).

3.2.2 Optical imaging

Microscopy optics is modeled here using two linear components: an effective point spread function (EPSF) and a lighting bias. DIC microscopy exploits the phase differential of the specimen to derive the edge representations of microscopy objects. The EPSF is approximated as a convolution of the OPL image $f(x, y)$ against the steerable first-derivative of Gaussian kernel, in (3.1).

Nonuniform lighting of a microscope often causes a pronounced lighting bias in the image. As presented in [96, 118], a linear approximation of quadratic coefficients sufficiently expresses such a bias:

$$b(x, y) = p_0 + p_1 x + p_2 y + p_3 x^2 + p_4 xy + p_5 y^2. \tag{3.6}$$

An example noiseless simulated image with the optical imaging model (including EPSF convolution and lighting bias) is shown in Fig. 3.2(c). Polynomials $p_0, ..., p_5$ are estimated from a randomly extracted patch from an actual DIC image using a least-squares framework detailed in [96].

3.2.3 Noise generation

Synthetic noise generation framework

Noise in a video frame may be linearly decomposed into organic and sensor components: $n_k(x, y) = n^{organic}(x, y) + n_k^{sensor}(x, y)$, with k denoting the frame index in time. We define $n^{organic}(x, y)$ as the components comprising of organic contributions in the specimen (e.g., cellular matter, fluids) that remain static frame-to-frame, while $n_k^{sensor}(x, y)$ refers to noise from the CCD that is iid across every pixel and frame. For the purpose of statistical estimation, we selected sequences of image frames $\{y_k(x, y)\}_{k=1,...,K}$ that represented noise only (i.e., no target cell) with no tissue motion from pipette insertion. We estimate the

organic noise component by averaging a series of image frames (to reduce sensor noise):

$$n^{organic}(x,y) \approx \bar{y}(x,y) = \frac{1}{K}\sum_{k=1}^{K} y_k(x,y).$$

Similarly, a sensor noise sample may be estimated by subtracting the the frame-average from the individual frame:

$$n_k^{sensor}(x,y) \approx y_k(x,y) - \bar{y}(x,y).$$

Organic Noise

We employ a spectral analysis framework to model and generate realistic looking interference caused by organic tissue. We define the radially averaged power spectral density (RAPSD), $P(f)$, as an averaging of the power spectral density (PSD) magnitudes along a concentric ring (whose radius is proportional to frequency, denoted by f) on a DC-centered spatial PSD Fourier plot. The RAPSD of random noise images reveal spectral characteristics shown in Fig. 3.4(d). Phase information is simulated by randomly sampling from a uniform distribution $\Phi(m,n) \sim \text{Uniform}([0,2\pi])$, with m,n representing the spatial coordinates in the DC-centered Fourier domain, while the magnitude information is composed as a mean spectra $\bar{P}(f)$ from RAPSD of example image patches $\{P_k\}_{k=1,\ldots,K}$ as

$$\bar{P}(f) = \frac{\sqrt{2}}{K}\sum_{k=1}^{K} P_k(f).$$

The organic noise spectrum is thus described by its magnitude and phase components as

$$\widetilde{S}(m,n) = |\bar{P}(\sqrt{m^2+n^2})| \exp(j \cdot \Phi(m,n)).$$

An inverse spatial Fourier transform on the organic noise spectrum yields a synthetic organic noise image: $\widetilde{n}^{organic}(x,y) = \mathcal{F}^{-1}\{\widetilde{S}(m,n)\}$. An example organic noise image

patch $\widetilde{n}^{organic}(x, y)$ (shown in Fig. 3.4(a)) exhibits visual similarities with a patch of organic noise from an actual DIC image (shown in Fig. 3.4(b)). Additionally, similarities were observed in pixel distributions between synthetic organic noise and actual organic noise, qualitatively as shown in Fig. 3.4(c), as well as quantitatively via a Kolmogorov-Smirnov two-sample goodness-of-fit test at the 5% significance level (after normalization by their respective sample standard-deviations).

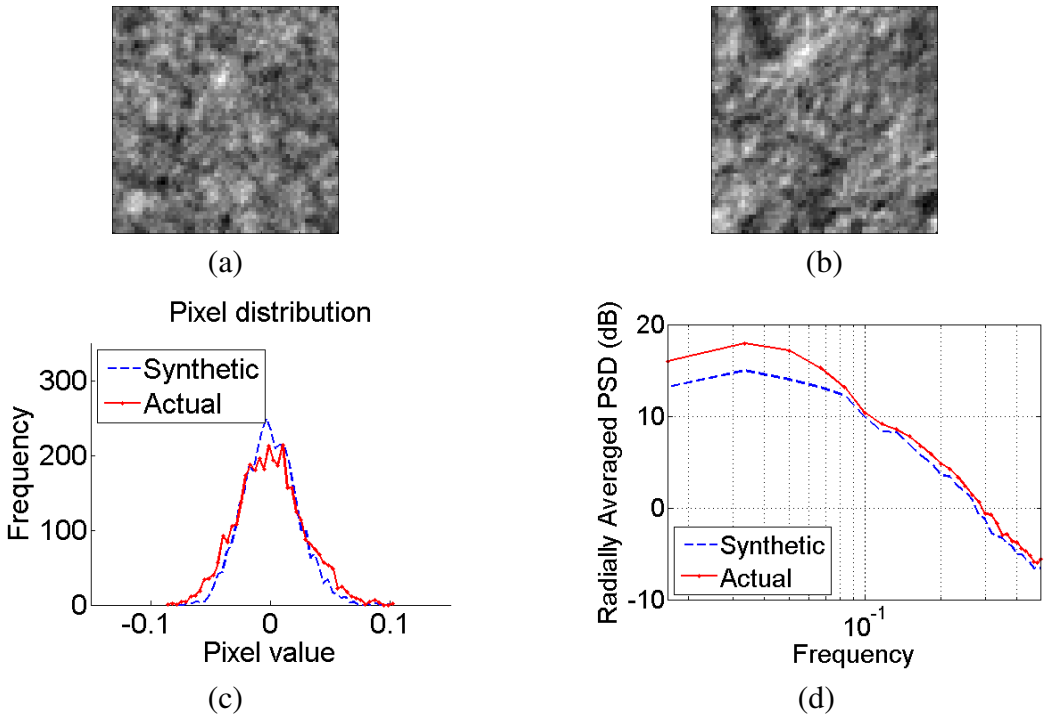


Figure 3.4: (a) Synthetically generated organic noise (synthetic) (b) An image patch of organic noise from a real DIC image (i.e. an image patch with no cell, only noise) (c) Comparisons of pixel intensity distributions (d) Comparisons of radially averaged power spectral density.

Sensor noise

The CCD sensor contributes a mixture of Poisson noise and zero-mean white-Gaussian noise [118, 116]. For modeling simplicity we assume that the sensor's individual pixels are

uncorrelated in time and space, and generated using

$$n_k^{sensor}(x,y) = A_g \cdot n_g + A_p \cdot n_p, \tag{3.7}$$

where $n_g \sim \mathrm{Normal}(0, \sigma_g^2)$ and $n_p \sim \mathrm{Poisson}(\lambda_p)$ are randomly generated intensity values, $\{A_\mathrm{g}, A_\mathrm{p}\}$ are amplitude parameters, and $\{\sigma_\mathrm{g}, \lambda_\mathrm{p}\}$ are the Gaussian's standard deviation and the Poisson's mean parameters respectively.

3.2.4 Image and Video Synthesis

Image synthesis

Each simulation frame is generated as a linear combination of the various synthesized components

$$y(x,y) = \left[10^{\chi/10}\frac{\|n(x,y)\|_F}{\|g(x,y)\|_F}\right] \cdot g(x,y) + n(x,y) + b(x,y), \tag{3.8}$$

where χ is a user-defined signal-to-noise ratio (in dB), and where $\|\cdot\|_F$ is the Frobenius norm. Table 3.1 summarizes the user-defined parameters of this simulator.

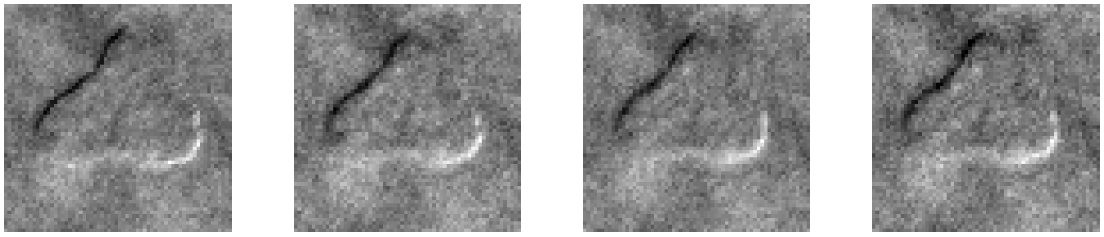
Video synthesis

During the patch clamp process, pipette motion causes the cells to undergo overall translation (e.g., moving from left to right with respect to the frame), and some sequence of dilation and contraction. Videos of K image frames are generated to simulate motion of the cell (rather than the pipette itself) by evolving a single textured OPL pixel-space embedding image over time using a series of geometric transformations. Specifically, we simulate a dialation/contraction using MATLAB's barrel transformation function [119] and apply geometric translation along a random linear path through the center of the image with a parabolic velocity profile (e.g., an acceleration followed by a deceleration). These transformations produced a set of frames $\{f_k(x,y)\}_{k=1,2,\ldots}$ which are convolved in 2D using (3.1) and synthesized using (3.8) to generate a set of time-varying observations $\{y_k(x,y)\}_{k=1,2,\ldots}$

Table 3.1: User-defined cell simulator parameters.

Parameter	Description
M	Image size (i.e., $M \times M$ pixel-image)
θ_{rot}	Rotation angle of cell-shape
γ_{scale}	Scaling factor of cell-shape within image
p	Persistence of OPL surface texture
$\{\sigma_{\text{d}}, \theta_{\text{d}}\}$	DIC imaging function parameters (3.1)
$\{A_{\text{g}}, \sigma_{\text{g}}\}$	Dynamic noise Gaussian parameters (3.7))
$\{A_{\text{p}}, \lambda_{\text{p}}\}$	Dynamic noise Poisson parameters (3.7))
χ	SNR (in dB) of final image (3.8)

comparable to a video sequence of DIC imagery from a patch clamp experiment. Example snapshots from a synthetic video is shown in Fig. 3.5.



$k = 1$ $k = 33$ $k = 66$ $k = 100$

Figure 3.5: Four snapshots in time (indexed by k) from a synthesized video (of 100 frames), generated from a single cell image. The cell's motion induced by external forces (i.e., pipette motion, though not explicitly present) is simulated by a slight contraction followed by expansion over time, while performing a linear translation, from left to right of the frame. Synthetic interference (simulating organic material) shows up as a high-intensity blob around the bottom left corner of the cell, interfering with the cell's edges.

3.2.5 Cell-shape realism evaluation

To evaluate the realism of synthetically generated cell-shapes, we compared several of their shape features to actual hand-drawn cell shapes from rodent brain slice imagery using four dimensionless shape features:

1. *Aspect Ratio* is defined as the ratio of minor axis lengths to major axis lengths. The major/minor axis is determined from the best fit ellipsoid of the binary image.

2. *Form factor* (sometimes known as circularity) is the degree to which the particle is similar to a circle (taking into consideration the smoothness of the perimeter [120]), defined as $\frac{4\pi A}{P^2}$, where A is area and P is the perimeter.

3. *Convexity* is a measurement of the particle edge roughness, defined as $\frac{P_{\text{cvx}}}{P}$, where P_{cvx} is the convex hull perimeter and P is the actual perimeter [121].

4. *Solidity* is the measurement of the overall concavity of a particle, defined as $\frac{A}{A_{\text{cvx}}}$, where A_{cvx} is the convex hull area and A is the image area [121].

Using 50 shape-coordinates per cell we generated 115 synthetic cells. The quartile-quartile plots in Fig. 3.6 show that all four features are similarly distributed, demonstrating that the simulation renders realistic cell shapes for rodent *in vitro* brain slices.

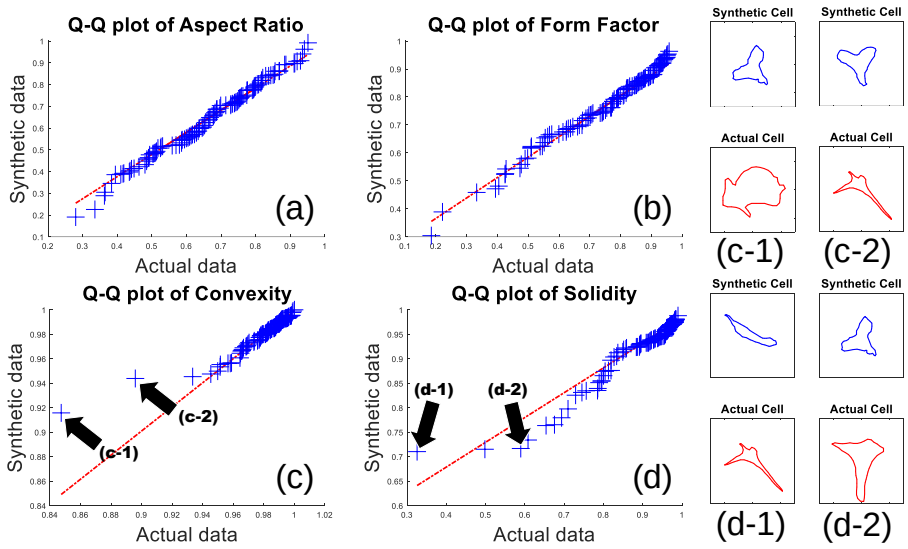
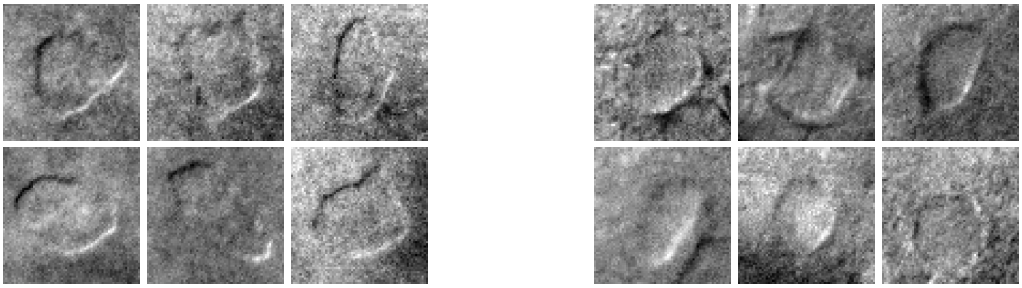


Figure 3.6: The quantile-quantile plots visualize similarities between the characteristics of simulated and actual cell-shapes. Distributions were compared for the following shape-features: (a) aspect ratio, (b) form factor, (c) convexity, and (d) solidity as defined in the text. (c-1,c-2,d-1,d-2) Several outlier pairs from the convexity and solidity plots are identified and shown. The outlier pairs appear to be caused by actual cell shape outlier statistics that are under-represented (due to lack of training examples) and thus not fully captured by the model.

Several simulation images were generated in Fig. 3.7 for visual comparison against actual cell images. The following user-defined parameters (as defined in Table 3.1 found

in Appendices) were used: image size was 60×60, cell-rotation (θ_{rot}) was randomized, cell-scaling defined as factor of image width was $\gamma_{\text{scale}} = 0.8$, DIC EPSF parameters were $\{\theta_{\text{d}} = 225°, \sigma_{\text{d}} = 0.5\}$, dynamic noise parameters were $\{A_{\text{g}} = 0.98, \sigma_{\text{g}} = 0.018, A_{\text{p}} = 20.6, \lambda_{\text{p}} = 10^{-10}\}$, and SNR $\chi = -1$ dB. Cell shapes were randomly generated based on learned real cell-shapes, a lighting bias was taken randomly from actual image patches, and organic noise was generated using a RAPSD curve learned from real DIC microscopy images. In general, the synthetic and actual cells are qualitatively similar in cell shapes and noise textures.



(a) (b)

Figure 3.7: General qualitative similarity between *randomly* drawn samples of (a) synthetically generated DIC cell images using the proposed simulator, and (b) actual DIC microscopy images of rodent neurons. Similarities in imagery characteristics include: (i) DIC optical features as shown by the 3D relief which are highlighted by the high/low intensity 'shadows' in the cell edges, (ii) organic interference which appears as Gaussian noise with a frequency profile (further elaboration on quantitative similarities are found in Section 3.2.3), (iii) cell shapes as demonstrated by the cell shapes' organic and natural contours, and (iv) uneven lighting bias as shown by the smooth but uneven background gradient.

3.3 Membrane tracking algorithm

The goal of the system is to provide automated visual tracking of the membrane of a user-selected target cell to guide a robotic patch clamping system. We will achieve these goals with an approach described by three general stages visualized by Fig. 3.8. The first stage uses standard computer vision tracking techniques to identify the general patch of interest containing the target cell on the current frame. The second stage implements dynamic

deconvolution to recover a time-varying OPL image that can be used for segmentation. The novel contribution of this section is a proposed deconvolution algorithm for this stage, and the details of this algorithm will be the focus of this section. The algorithm contains two distinct components: a pre-filtering (PF) operation, and the re-weighted total variation dynamic filtering (RWTV-DF) deconvolution algorithm. The two parts play distinct yet important roles: PF performs interference suppression while the RWTV-DF deconvolution achieves dynamic edge-sparse reconstructions. We will also describe an extension of the basic proposed algorithm that retains good performance when the recording pipette overlaps with the cell by removing the pipette image and performing inpainting. The third stage is a segmentation on the output of the deconvolution, which is performed with simple thresholding. We point out that a simple segmentation strategy is sufficient after a robust deconvolution process.

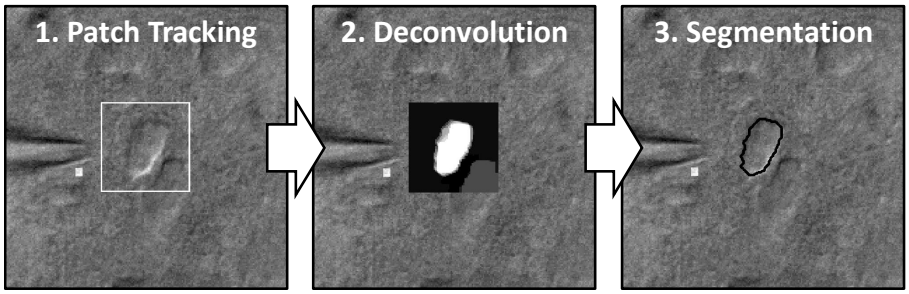


Figure 3.8: The full required imaging system has three stages: patch tracking, deconvolution, and segmentation. In patch tracking, a user-defined template is provided for template matching and tracking spatial coordinates over time with a Kalman filter. In deconvolution (the main focus of this section), the patch is deconvolved to infer the OPL approximation. In segmentation, a global threshold is applied (using Ostu's method [122]) to yield a binary segmentation mask to determine the cell membrane location.

3.3.1 Problem setup

The deconvolution and segmentation algorithms will be run on a limited image patch size (e.g., 64×64, as a trade-off between speed and resolution) that is found by tracking gross motion in the image. Specifically, template matching (via normalized cross-correlation)

is performed on each frame using a user-selected patch (e.g., obtained via a mouse-click to associate coordinates of the image patch center with the targeted cell's center). Patch-tracking robustness may be improved by running a standard Kalman filter on the positional coordinates of the found patch (i.e., tracking a two-dimensional state vector of the horizontal and vertical location of the patch center). To ensure efficacy with the correlation approach, a preprocessing stage of lighting bias elimination (i.e., uneven background subtraction) is performed on each video frame[2] using quadratic least squares estimation (with a polynomial order of 2) as described in [96] and by (3.6). While tracking the detailed cell membrane locations is challenging, tracking the general location of the patch containing the target cell can be done with very high accuracy using this approach. Yet, more sophisticated and robust computer vision methods such as multi-cue visual tracking [123, 124, 125, 126] can replace this basic correlation approach to improve on the patch-tracker's performance. Specific implementation details to the template matching approach may be found in [89].

Following insight gained from developing the simulator in section 3.2, we assume that at each time step (indexed by k) the bias eliminated image has the following generative model:

$$\boldsymbol{y}_k = \boldsymbol{D}\boldsymbol{x}_k + \boldsymbol{n}_k^{organic} + \boldsymbol{n}_k^{sensor},$$

where $\boldsymbol{y}_k \in \mathbb{R}^N$ is the vectorized form of the microscopy image (e.g., 64×64 image patch), $\boldsymbol{n}_k^{organic} \in \mathbb{R}^N$ is a noise component consisting of organic materials surrounding the cell and modeled by a non-white Gaussian distribution (i.e., non-flat frequency profile), and $\boldsymbol{n}_k^{sensor} \in \mathbb{R}^N$ consists of 2 distinct additive noise elements: thermal noise modeled as a white-Gaussian distribution and photon noise modeled as a Poisson distribution. We also

[2]Bias elimination was applied to the entire video frame (rather than to each patch) for computational efficiency. Also, if the bias conditions are static, it could be computed once, cached, then applied across multiple frames.

assume that the signal evolution over time is governed by a dynamics model:

$$\boldsymbol{x}_k = g_k(\boldsymbol{x}_{k-1}) + \boldsymbol{\nu}_k, \tag{3.9}$$

where $g_k(\cdot) : \mathbb{R}^N \to \mathbb{R}^N$ is a dynamic evolution function and $\boldsymbol{\nu}_k \in \mathbb{R}^N$ is an innovations term representing errors in the dynamic model $g_k(\cdot)$.

We note that DIC deconvolution is particularly sensitive to uneven (i.e., non-matching) boundaries. Hence, the implementation of the matrix operator $\boldsymbol{D}$, though application dependent, should be treated carefully. We employ a discrete convolutional matrix implementation that handles non-matching boundary but could cause memory issues when N is large since $\boldsymbol{D}$ scales quadratically in size (though it could be mitigated using sparse matrices). Alternatively, an FFT/IFFT surrogate implementation implies circular boundary conditions that need to be explicitly taken care of (e.g., via zero-padding).

3.3.2 Pre-filtering for interference suppression

The presence of non-white Gaussian noise due to organic tissue interference causes significant challenges in recognizing membrane boundaries. The pre-filter's goal is to estimate the interference-free observation, which is achieved by "whitening" the spectra associated with the observation and amplifying the spectra associated with the signal. Specifically, the filter's design is derived using the Wiener filter and given in the spatial-frequency domain by:

$$|F_k|^2 = \frac{|\mathcal{F}\{d\} \cdot \widehat{X}_k|^2}{|\mathcal{F}\{d\} \cdot \widehat{X}_k|^2 + |\widehat{N}_k|^2} = \frac{|\mathcal{F}\{d\} \cdot \widehat{X}_k|^2}{|\widehat{Y}_k|^2}, \tag{3.10}$$

where F_k is the spatial Fourier spectrum of the pre-filter, $\mathcal{F}\{d\}$ is the spatial Fourier transform of the DIC imaging function from (3.1), $\widehat{X}_k$ is the OPL signal's spatial Fourier spectrum estimate, $\widehat{N}_k$ is the spatial Fourier spectrum estimate of $\boldsymbol{n}_k^{organic}$ combined with $\boldsymbol{n}_k^{sensor}$, and $\widehat{Y}_k$ is the signal-plus-noise's spatial Fourier spectrum estimate. We estimate all spatial Fourier spectra using least-squares polynomial fits of the radially averaged power

spectral density (RAPSD), defined as a radial averaging of the DC-centered spatial-frequency power spectrum density. Intuitively, this estimates an image's underlying spectrum via a form of direction-unbiased smoothing. Specifically, we estimate the signal-only component $\widehat{X}_k$ from averaged RAPSDs from OPLs of simulated cells generated from the realistic simulation framework described in Section 3.2, and we estimate the combined signal-plus-noise spectra $\widehat{Y}_k$ from RAPSDs of observed images $\boldsymbol{y}_k$. Fig. 3.9 illustrates the estimation process and effect of the pre-filter in the frequency domain. The positive contribution of the pre-filter is highlighted in Fig. 3.10 in the segmentation algorithm to be described next.

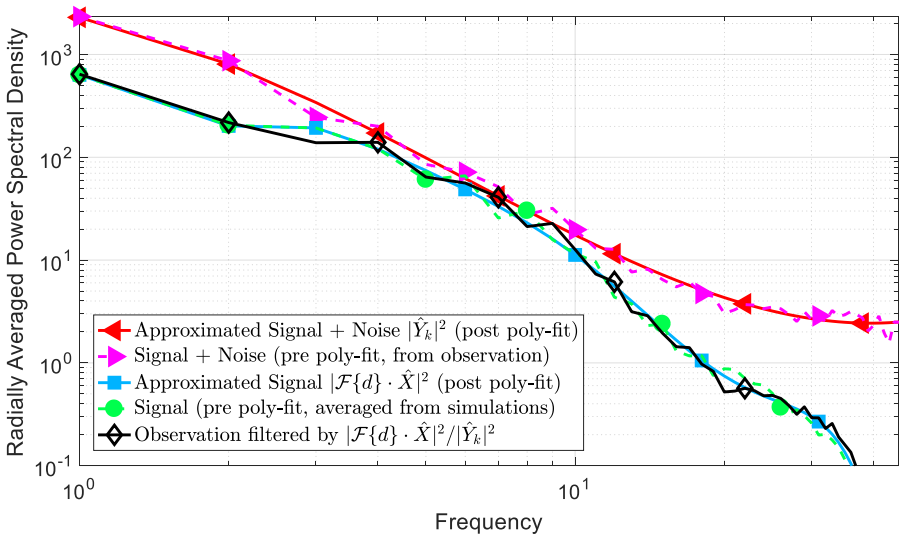


Figure 3.9: Radially averaged power density spectrum (RAPSD) plots in log-scale from an example 64×64 cell patch. The Wiener filter utilizes an approximation to the signal (in blue) and the signal-with-noise (in red) power spectral density modeled as a linear polynomial least-square fitting over the radially averaged power spectral density (RAPSD). The signal approximation is computed by averaging the power spectral density of simulated cells. The signal-with-noise approximation is extracted from an observed patch. The final filtered spectrum is shown in black.

3.3.3 Reweighted total-variation dynamical filtering deconvolution

Given the edge-sparse nature of the data and our particular need for accuracy in the cell membrane locations during segmentation (in contrast to the more typical MSE minimiza-

tion of the deconvolved image), we use the TV norm as the core regularization approach [16] for this algorithm. Specifically, we calculate isotropic TV as:

$$\|\boldsymbol{x}_k\|_{\text{TV}} = \sum_{i=1}^{N} \|\boldsymbol{T}_i \boldsymbol{x}_k\|_2 = \sum_{i=1}^{N} \boldsymbol{t}_k[i],$$

where $\boldsymbol{T}_i \in \mathbb{R}^{2\times N}$ is an operator that extracts the horizontal and vertical forward-differences of the i-th pixel of $\boldsymbol{x}_k$ into an $\mathbb{R}^2$ vector, whose ℓ_2 norm we denote as an individual edge-pixel $\boldsymbol{t}_k[i]$. This basic regularizer was improved in [13] via an iterative estimation of weights on the individual edges through an Majorize-Minimization algorithm. While effective, this regularizer does not incorporate any dynamical information for the tracking of moving edges.

To address this issue, we draw inspiration from [15, 14] and develop a hierarchical Bayesian probabilistic model where dynamics are propagated via second-order statistics from one time step to the next, the key difference being our proposed method operates solely in the edge-pixel space. To give intuition first, we note that the sparse edge locations at the previous frame is strong evidence that there will be an edge nearby in the current frame. We model edge locations with a sparsity-inducing probability distribution with a parameter controlling the spread (i.e., variance) of the distribution. When previous data gives evidence for an edge in a given location, we adjust the parameter to increase the variance of the prior in this location, thereby making it easier for the inference to identify the presence of the edge from limited observations. In contrast, when previous data indicates that an edge in a location is unlikely, this variance is decreased thereby requiring more evidence from the observations to infer the presence of an edge. For more details about this general approach to dynamic filtering (including its enhanced robustness to model uncertainty) see [15].

In detail, at the lowest level of the hierarchy, we model the pre-filtered observations

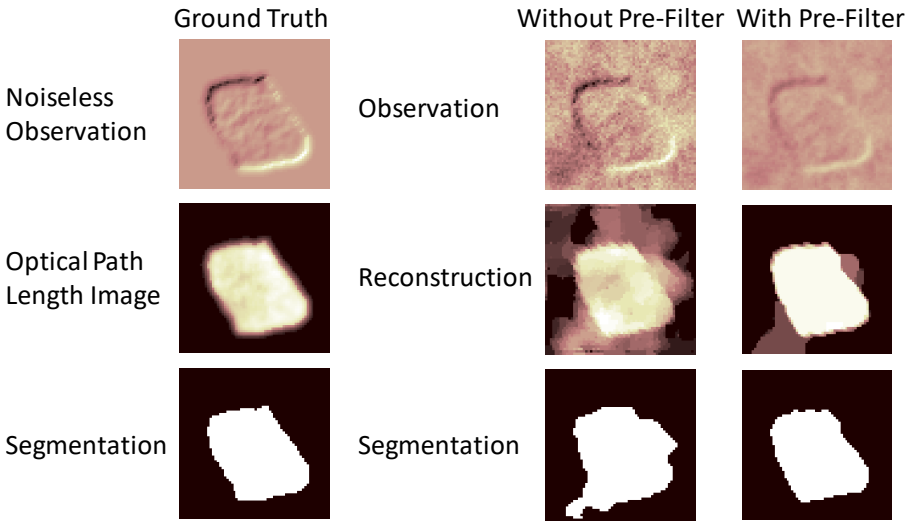


Figure 3.10: Positive contribution of pre-filtering seen as interference-suppression. The first column of images pertain to the ground truth simulation. The second and third columns pertain to PF+RWTV-DF performed without and with the pre-filter respectively. While the pre-filter blurs high frequency interference content, it also blurs the cell's edges. We consider this an acceptable trade-off that we can thereafter comfortably recover from, using our proposed deconvolution method. The pre-filtered observation has its edge integrity retained while suppressing surrounding interference found at the bottom left corner and immediate right of the cell; this results in an overall segmentation that is closer to the ground truth.

conditioned on the OPL signal $\boldsymbol{x}_k$ with a white Gaussian distribution:

$$p(\boldsymbol{F}_k\boldsymbol{y}_k|\boldsymbol{x}_k) \propto \exp\Big(-\frac{1}{2\sigma^2}\|\boldsymbol{F}_k\boldsymbol{y}_k - \boldsymbol{D}\boldsymbol{x}_k\|_2^2\Big), \tag{3.11}$$

where $\boldsymbol{F}_k$ is the matrix operator describing 2D convolution (in the frequency domain) against the pre-filter F_k described earlier by (3.10). At the next level, the individual edge-pixels $\boldsymbol{t}_k[i]$ are assumed sparse and therefore best modeled with a distribution with high kurtosis but with unknown variance. Conditioning on a weighting $\boldsymbol{\gamma}_k[i]$ that controls the individual variances, we model $\boldsymbol{t}_k[i]$ as random variables arising from independent Laplacian distributions:

$$p(\boldsymbol{t}_k[i]|\boldsymbol{\gamma}_k[i]) = \gamma_0\frac{\boldsymbol{\gamma}_k[i]}{2}\exp(-\gamma_0\boldsymbol{\gamma}_k[i]\cdot|\boldsymbol{t}_k[i]|), \tag{3.12}$$

where γ_0 is a positive constant. At the top-most level, the weights $\boldsymbol{\gamma}_k[i]$ are themselves also

treated as random variables with a Gamma hyperprior:

$$p(\boldsymbol{\gamma}_k[i]|\boldsymbol{\theta}_k[i]) = \frac{\boldsymbol{\gamma}_k^{\alpha-1}[i]}{\boldsymbol{\theta}_k^{\alpha}[i]\Gamma(\alpha)} \exp(-\boldsymbol{\gamma}_k[i]/\boldsymbol{\theta}_k[i]), \tag{3.13}$$

where α is a positive constant, $\Gamma(\cdot)$ is the Gamma function, and $\boldsymbol{\theta}_k[i]$ is the scale variable that controls the Gamma distribution's mean and variance over $\boldsymbol{\gamma}_k[i]$. We will use this top level variable to insert a dynamics model into the inference that controls the variance of the prior being used to infer edge locations based on previous observations. Fig. 3.11 illustrates the multiple prior dependencies of the hierarchical Laplacian scale mixture model described in (3.11)-(3.13) using a graphical model.

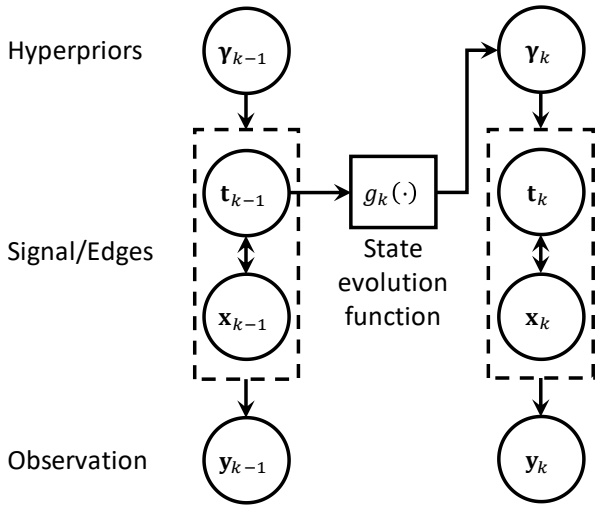


Figure 3.11: Graphical model depicting the hierarchical Laplacian scale mixture model's Bayesian prior dependencies in the RWTV-DF algorithm. Prior state estimates of the signal edges are used to set the hyper-priors for the second level variables (i.e., variances of the state estimates), thereby implementing a dynamical filter that incorporates edge information into the next time step.

To build dynamics into the model, we observe that the Gamma distribution's scale variable $\boldsymbol{\theta}_k[i]$ controls its expected value (i.e., $\mathbb{E}\big[\boldsymbol{\gamma}_k[i]\big] = \alpha\boldsymbol{\theta}_k[i]$). Inspired by [15], we designed a dynamic filtering approach to the problem of interest by using a dynamics model on the edge-pixels that sets the individual variances $\boldsymbol{\theta}_k[i]$'s according to predictions from

the dynamics model $g_k(\cdot)$:

$$\theta_k[i] = \frac{\xi}{|g_k(\boldsymbol{t}_{k-1})[i]| + \eta},$$

where ξ is a positive constant and η is a small constant that prevents division by zero. To illustrate the operation of this model, a strong edge-pixel in a previous frame (i.e., a large value of $\boldsymbol{t}_{k-1}[i]$) sets a small value of $\boldsymbol{\theta}_k[i]$ (and respectively a small expected value of $\boldsymbol{\gamma}_k[i]$), in turn making the Laplacian's variance large (i.e., $\text{Var}\big[\boldsymbol{t}_k[i]\big] = 2/(\gamma_0 \boldsymbol{\gamma}_k[i])^2$). A large Laplacian variance implies a higher likelihood that the edge-pixel in the current frame is active (in contrast to the reverse situation where a weak edge pixel would result in a small Laplacian variance at the next time step). Therefore, this approach propagates second-order statistics (similar to classic Kalman filtering for Gaussian models) through the hyper-priors $\boldsymbol{\gamma}_k$ using dynamic information via the evolution function $g_k(\cdot)$ at each time-step. One option for $g_k(\cdot)$ is a convolution against a Gaussian kernel (with its σ proportional to expected motion variation), which expresses a confidence neighborhood of edge locations based on previous edge locations [127]. As with any tracking algorithm, the tracking quality depends on the accuracy of the dynamics model and including better models (e.g., more accurate motion speeds, motion direction information based on pipette movement, etc.) would improve the performance of any approach. With a fixed dynamics function, for each frame the algorithm will take a re-weighting approach where multiple iterations are used to adaptively refine the estimates at each model stage (illustrated for a simulated patch in Fig. 3.12).

The final optimization follows from taking the MAP estimate for (3.11)-(3.13) and applying the EM approach to iteratively update the weights. The maximization step is given as a convex formulation

$$\begin{aligned} \widehat{\boldsymbol{x}}_k^{(t)} &= \underset{\boldsymbol{x} \geq \boldsymbol{0}}{\arg\min} \;\; -\log\Big[p(\boldsymbol{x}|\boldsymbol{\gamma}_k^{(t)})\Big] \\ &= \underset{\boldsymbol{x} \geq \boldsymbol{0}}{\arg\min} \;\; \frac{1}{2}\left\|\boldsymbol{F}_k \boldsymbol{y}_k - \boldsymbol{D}\boldsymbol{x}\right\|_2^2 + \gamma_0 \sum_{i=1}^{N} \boldsymbol{\gamma}_k^{(t)}[i] \cdot \left\|\boldsymbol{T}_i \boldsymbol{x}_k\right\|_2 . \end{aligned} \tag{3.14}$$

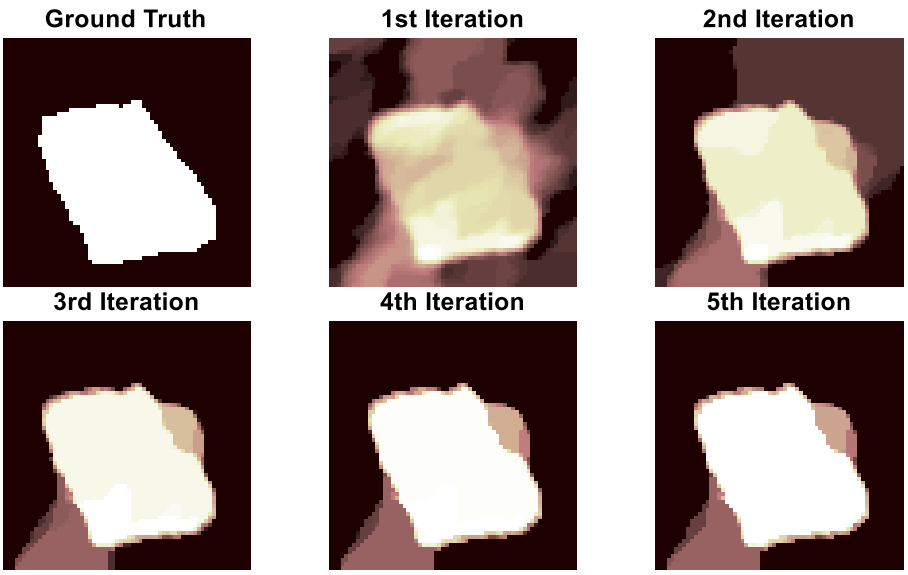


Figure 3.12: The ground truth image is the binary mask of a simulated image. Subsequent images reflect the iterations of reweighting process during deconvolution (iterations 1 through 5 respectively). Over the iterations, dominant edges are enhanced while weaker ones recede, resulting in piecewise smooth solutions that are amenable to segmentation.

The expectation step may be derived using the conjugacy of the Gamma and Laplace distributions, admitting the closed-form solution

$$\boldsymbol{\gamma}_k^{(t+1)}[i] = \mathbb{E}_{p(\gamma|\widehat{\boldsymbol{t}}_k^{(t)}[i])}[\gamma] = \frac{\kappa + 1}{\kappa \cdot |\widehat{\boldsymbol{t}}_k^{(t)}[i]| + |g_k(\widehat{\boldsymbol{t}}_{k-1})[i]| + \eta}, \tag{3.15}$$

where the EM iteration number is denoted by superscript t, $\widehat{\boldsymbol{t}}_k^{(t)}[i] = \|\boldsymbol{T}_i \widehat{\boldsymbol{x}}_k^{(t)}\|_2$ is the estimate of the edge at iterate t, $\widehat{\boldsymbol{t}}_{k-1}[i] = \|\boldsymbol{T}_i \widehat{\boldsymbol{x}}_{k-1}\|_2$ is the estimate of the edge at the previous time-step, γ_0 is the positive constant that controls the weight of edge-sparsity against reconstruction fidelity, and κ is the positive constant that controls the weight of the current observation against the dynamics prior. The Gaussian convolution implemented by $g_k(\cdot)$ effectively "smears" the previous time-step's estimation of edges to form the current prior, accounting for uncertainty due to cell movement over time. The EM algorithm is initialized by setting all weights to 1 (i.e., $\boldsymbol{\gamma}_k^{(0)} = \mathbf{1}$) and solving (3.14), which is simply (non-weighted) TV-regularized least squares. The reweighting iterations are terminated upon

reaching some convergence criteria (e.g., $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)}\|_2 / \|\boldsymbol{x}^{(t)}\|_2 < \varepsilon$ or after a fixed number of reweights).

3.3.4 Simulations

We test the PF+RWTV-DF algorithm on synthetic cell data and compare its performance against the state-of-the-art deconvolution algorithms such as least-squares regularized by ℓ_1 and TV (L1+TV) [96], least-square regularized by ℓ_1 and Laplacian Tikhonov (L1+Tik) [96], and least-square by regularized re-weighted ℓ_1, weighted Laplacian Tikhonov, and weighted dynamic filtering (RWL1+WTik+WDF) [107]. The deconvolution from each algorithm is then segmented using a global threshold (using Otsu's method [122]) to produce a binary image mask for algorithmic evaluation.

For evaluation, we employ three boundary metrics that capture errors relevant to the process of patch clamping. The metrics *Average Boundary Error* (ABE), *Maximum Boundary Error* (MBE), and *Variance of Boundary Errors* (VBE) measure the (average, maximum, and variance of) distance errors between the approximated cell boundary and the actual cell boundary around the entire membrane of the cell at each video frame. Specifically, these metrics capture the statistics of the distance errors between the actual and estimated cell boundaries as the pipette tip approaches the cell membrane in a straight path (directed towards the true centroid).

To compute these metrics, we first discretized potential pipette paths towards the true centroid using a set of lines $\{l_a\}$, defined as lines that start at the true centroid and that infinitely extend through each point p_a in the set of ground truth contour pixels $\{A\}$. Next, traveling along each l_a toward the centroid we found the intersecting point in the set of estimated contour pixels $\{B\}$, defined as $p_{a,B} = \arg\min_{p_b \in B} \|p_b - p_l\|_2$ s.t. $p_l \in l_a$, where p_l are points on the line l_a. In other words, we found the points that the algorithm would identify as the membrane location during every hypothetical approach direction of the pipette. The $\|p_b - p_l\|_2$ in the objective considers only orthogonal projections of p_b unto the line

l_a (since it is a minimization), and it searches over all of B to find the p_b whose orthogonal projection distance is minimal. Note that that due to the contour discretization, $p_{a,B}$ does not lie exactly along l_a but rather close to it on some $p_b \in B$. With each discretized boundary error expressed as $\varepsilon(a, B) = \|p_{a,B} - p_a\|_2$, the metrics are then defined as:

$$\text{ABE}(A, B) = \frac{1}{\|A\|_0} \sum_{a \in A} \varepsilon(a, B)$$

$$\text{MBE}(A, B) = \max_{a \in A} \varepsilon(a, B)$$

$$\text{VBE}(A, B) = \frac{1}{\|A\|_0 - 1} \sum_{a \in A} \Big(\varepsilon(a, B) - \text{ABE}(A, B) \Big)^2.$$

Fig. 3.13 illustrates the terms involved in these error metric computations.

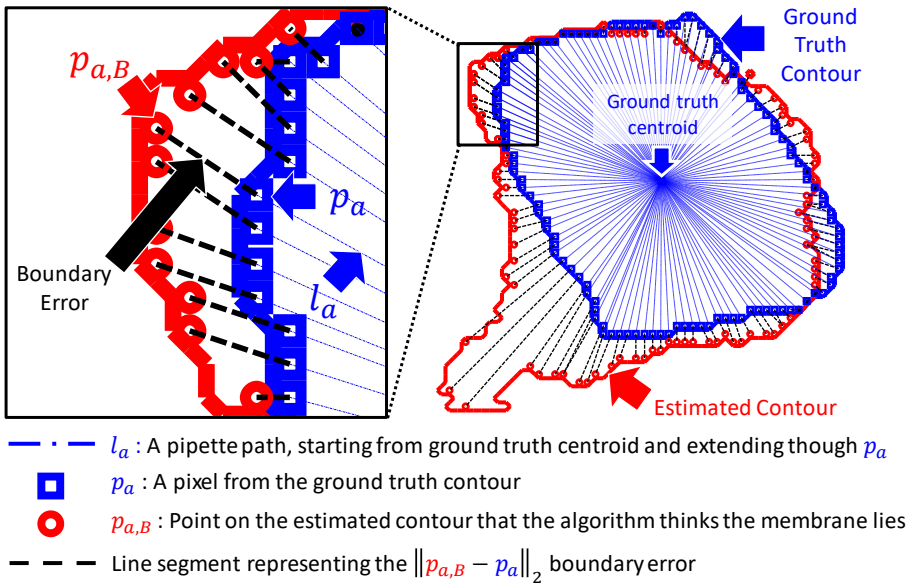


Figure 3.13: Illustration of boundary error metrics. Each blue line l_a represents an approach direction of a pipette towards the ground truth centroid. $p_{a,B}$ represents the point where the algorithm thinks the membrane is. Individual boundary errors are computed as the distance $\|p_{a,B} - p_a\|_2$ and represented as black dotted lines. The average of all errors constitutes the Average Boundary Error (ABE), the maximum constitutes the Maximum Boundary Error (MBE), and the variance of errors constitutes the Variance of Boundary Errors (VBE).

We generated 100 videos of synthetic cell patches (each video contains 64×64 pixels $\times$ 100 frames) using the proposed cell simulator with DIC EPSF parameters $\sigma_d = 0.5$

and $\theta_d = 225^\circ$ (assumed to be known either by visual inspection (to accuracy of $\pm 15^\circ$) or more accurately by using a calibration bead [99]). By physically relating real cells to simulated cells, we determined the relationship between physical length and pixels to be approximately 3.75 pixels per $1.0\mu m$.

For the algorithms L1+TV (3.2), L1+Tik (3.3), and RWL1+WTik+WDF (3.4), the parameters for sparsity (β) and smoothness (γ) were tuned via exhaustive parameter sweep on the first frame of the video (to find the closest reconstruction/deconvolution by the ℓ_2 least squares sense). Specifically, the sweep was conducted over 10 logarithmically spaced values in the given ranges: L1+TV searched over $[10^{-3}, 10^{-9}]$ for β and $[10^{-1}, 10^{-5}]$ for γ, L1+Tik searched over $[10^{-3}, 10^{-9}]$ for β and $[10^{-1}, 10^{-7}]$ for γ, and RWL1+WTik+WDF searched over $[10^{-3}, 10^{-9}]$ for β and $[10^{-1}, 10^{-5}]$ for γ. L1+Tik (the non-reweighted version) was used in all experiments because it was experimentally found to be better performing compared to the reweighted version (i.e., set $\mathbf{\Lambda} = \boldsymbol{I}$ for (3.3); reweighting over-induces sparsity of pixels, which is not advantageous to segmentation in our particular application. For RWL1+WTik+WDF, the dynamics parameter was set to be $\kappa = 1.0 \times 10^{-3}$, with a maximum of 80 reweighting steps (where convergence is fulfilled before each reweighting). For PF+RWTV-DF (3.14)-(3.15), we fixed $\gamma_0 = 3.0 \times 10^{-4}, \kappa = 5$, with 4 reweighting steps (where convergence is fulfilled before each reweighting). The programs specified by L1+TV, RWL1+WTik+WDF, and PF+RWTV-DF were solved using CVX [128, 129], while L1+Tik was solved using TFOCS [130].

Fig. 3.14 reflects statistics of the respective algorithms from each of the 100 frames from the 100 video trials. The height of each bar refers to the average of ABE/MBE/VBE while the error bars refer to ± 1 standard error of the ABE/MBE/VBE over the 100 video trials. PF+RWTV-DF was the best performing algorithm in the three metrics, and there is a significant difference ($p \leq 0.001$) between PF+RWTV-DF and the other algorithms for all metrics. Notably, PF+RWTV-DF results in boundary tracking errors on the order of 1-2 μm, which is comparable to the error in mechanical pipette actuators.

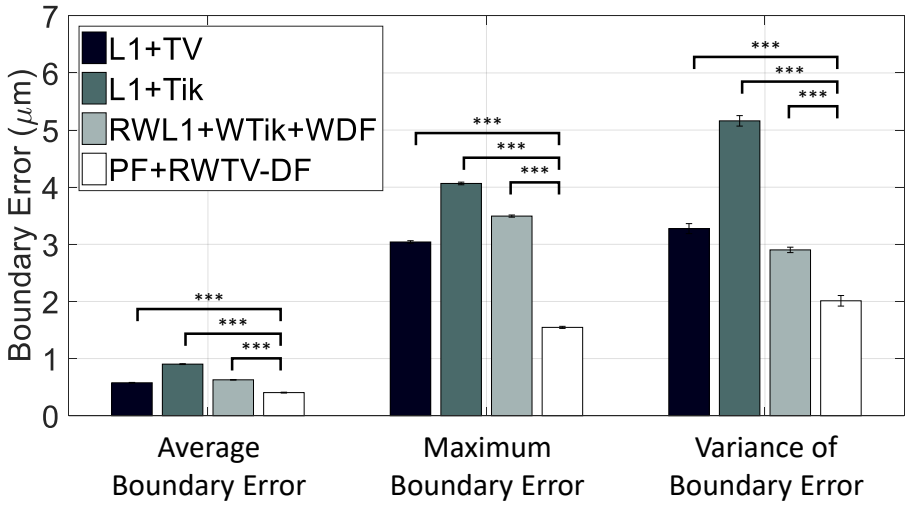


Figure 3.14: Average boundary errors (ABE), maximum boundary errors (MBE), and variance of boundary errors (VBE) for each of the algorithms across 100 video trials of simulated data. ABE/MBE/VBE was aggregated from 100 frames of 100 trials (i.e., 10,000 data points). The PF+RWTV-DF algorithm was found to have a state-of-the-art performance, showing statistically significant improvements in ABE/MBE/VBE compared to other algorithms ($p \leq 0.001$, denoted by the 3 stars based on paired t-tests).

The proposed edge reweighing strategy is adaptive by design, making it very attractive in practice because it requires minimal parameter fine-tuning. The algorithm's performance was responsive to the smoothness parameter (effective range being $\gamma \in [2.0 \times 10^{-4}, 4.0 \times 10^{-4}]$) and the dynamics parameter (effective range being $\kappa \in [10^0, 10^2]$), yet not oversensitive: our algorithm's superior results were from *fixed* parameters (γ, κ) across *all* trials, while other algorithms required exhaustive two-parameter sweeps (β, γ) for *each* synthetic video trial.

In Fig. 3.15 we take a closer look at one representative trial (Trial #4) in the time-series to observe the qualitative differences between the algorithms. We observe that the PF+RWTV-DF reconstruction has a fairly flat magnitude (for pixel values within the cell) compared to the other reconstructions. The proposed algorithm's inherent segmentation capability under such high-interference synthetic data is apparent from these results. Although PF+RWTV-DF reconstruction loses cell details (i.e., in an ℓ_2 reconstruction sense),

the reduction in surrounding interference serves to improve the boundary identification at the segmentation stage. On the other hand, the other algorithms produce reconstructions that contain significant surrounding interference, resulting in distorted edges in their subsequent segmentations (that may require further image processing to remove). Video showing the deconvolution and tracking time series is included in a multimedia supplement to our paper [91].

3.3.5 Pipette removal via inpainting

When the pipette draws near to a cell for patching, the pipette tip in the image patch can cause significant errors in the deconvolution process. We propose a simple yet effective extension of the proposed PF+RWTV-DF algorithm that requires only a minor modification of (3.14). Specifically, we propose an inpainting approach [131] that masks away the pipette's pixels and infers the missing pixels according to the same inverse process used for deconvolution. To begin, we track the pipette location in the image with a similar template matching process as described in Section 3.3.1 (which could be improved with positional information from the actuator if available). Simultaneously, an associated pipette mask (obtained either via automatic segmentation or manually drawn) is aligned and overlaid on top the cell image patch using the updated pipette locations. This pipette mask overlay takes the form of a masking matrix $\boldsymbol{M} \in \{0,1\}^{M \times N}$ whose rows are a subset of the rows of an identity matrix, and whose subset is defined by pixel indices outside the pipette region. The original filtered observation $\boldsymbol{F}_k \boldsymbol{y} \in \mathbb{R}^N$ is reduced to $\boldsymbol{M}\boldsymbol{F}_k \boldsymbol{y} \in \mathbb{R}^M$ (with $M \leq N$) while the convolution operator becomes $\boldsymbol{M}\boldsymbol{D}$. The inpainting version of (3.14) is

$$\widehat{\boldsymbol{x}}_k^{(t)} = \underset{\boldsymbol{x} \geq \boldsymbol{0}}{\arg\min} \ \frac{1}{2} \left\| \boldsymbol{M}\boldsymbol{F}_k \boldsymbol{y}_k - \boldsymbol{M}\boldsymbol{D}\boldsymbol{x} \right\|_2^2 + \gamma_0 \sum_{i=1}^{N} \gamma_k^{(t)}[i] \left\| \boldsymbol{T}_i \boldsymbol{x} \right\|_2 . \tag{3.16}$$

Intuitively, we simply remove pixels in the pipette region from the data used in the deconvolution but allow the algorithm to infer pixel values consistent with the statistical model.

3.3.6 Qualitative real data results

In addition to the synthetic data where we have known ground truth, we also tested the algorithms on *in vitro* DIC microscopy imagery of rodent brain slices from the setup described in [113]. For each algorithm, a parameter search was performed using a brute-force search to find the parameters that best segmented the cell by means of visual judgment. The DIC EPSF was estimated visually to be $\sigma_d = 0.5, \theta_d = 225^\circ$.

In Fig. 3.16, we selected 3 particularly challenging cell samples with respect to the amount of observable interference around and within the cell. A fixed global threshold was applied to each patch for segmentation using Otsu's method [122]. The first example (from the top) demonstrates that reconstructions of other algorithms (compared to the proposed algorithm) are characteristically *not* piecewise smooth. Therefore, even in a case of moderate difficulty such as this one, these algorithms produce images with rounded edges which are not ideal for segmentation. The second example illustrates difficulty in segmenting cells with significant interference along the cell's edges, around the outside of the cell, and within the body of the cell. In this case, only the proposed algorithm is able to produce a clean segmentation, especially along the cell's edges. Moreover, other algorithms reproduce the heavy interference scattered around the outside of the cell and this requires subsequent image processing to remove. The third example is a very difficult case where interference occurs not only as distorted boundaries, but also as a close neighboring cell. All other algorithms visibly perform poorly while the proposed algorithm remains fairly consistent in its performance.

In Fig. 3.17, snapshots from a video of a cell undergoing patch clamping was compared with and without the proposed pipette removal method (from section 3.3.5). The pipette (and its respective mask) was tracked using a template matching algorithm via a user-selected template of the pipette; more details are described in [89]. In Fig. 3.17(a), the pipette does not cause interference when it is relatively far away from the cell. As the pipette approaches the cell in Fig. 3.17(b), interference begins to enter the deconvolution

when inpainting is not applied. Without inpainting, an obfuscation between the pipette and the cell is observed in the deconvolution even when the pipette is not overlapping with the cell. In Fig. 3.17(c), the pipette is now seen to be overlapping the cell image. Without inpainting, this causes such severe interference that attempting a pipette removal at the segmentation stage is clearly non-trivial. With inpainting, the deconvolution is shown to effectively suppress the interfering pipette in all three snapshots.

3.4 Real-time implementation on parallel hardware

In this section, we seek a real-time implementation for just the RWTV part since this is the computational bottleneck, and it generalizes the full algorithm (3.16)-(3.15). Stated explicitly, we wish to efficiently solve the following iterative algorithm:

$$\widehat{\boldsymbol{x}}^{(k+1)} = \underset{\boldsymbol{x}\geq\boldsymbol{0}}{\arg\min}\ \frac{1}{2}\left\|\boldsymbol{y}-\boldsymbol{M}\boldsymbol{A}\boldsymbol{x}\right\|_2^2 + \gamma_0 \sum_{i=1}^{N} \gamma_i^{(k)} \left\|\boldsymbol{T}_i\boldsymbol{x}\right\|_2, \tag{3.17}$$

$$\gamma_i^{(k+1)} = \frac{1}{\left\|\boldsymbol{T}_i\widehat{\boldsymbol{x}}^{(k)}\right\|_2 + \epsilon}. \tag{3.18}$$

A major pitfall with this method is that it is expensive because each reweight requires the optimization program (3.17) to be solved. To mitigate this, we seek fast methods of solving the optimization. The alternating direction method of multipliers (ADMM) has reemerged in recent times as an efficient optimization tool [22] that utilizes a variable splitting approach to decompose the optimization into subproblems, that are much cheaper to solve than the original problem. ADMM frameworks that have been proposed to solve deconvolution and inpainting problems [132, 133, 134, 135, 31] require, at its crux, a matrix inversion at each ADMM iteration. Fortuitously, this is efficiently performed using the fast Fourier transform (FFT) since convolution may be generally assumed by periodic/circular matrices.

We build on the ADMM framework by Almeida and Figueiredo for (unweighted) TV deconvolution and inpainting [31] and extend it to the *reweighted* TV paradigm. We

present a simple yet effective accelerated reweighing framework for this purpose, combining strengths of reweighted sparsity approaches and efficiency of ADMM. In addition, we describe and demonstrate a GPU-accelerated implementation of the proposed algorithm for a deconvolution and inpainting problem in differential interference contrast (DIC) microscopy for neuronal-cell segmentation.

We adopt the general reweighted sparsity approach that requires two nested loops: an inner-loop that performs the optimization, i.e., Eq. (3.17), and an outer-loop that performs reweighing, e.g., Eq. (3.18). To accelerate the reweighing, we compute weights from only partially converged solutions, followed by warm-starting each new reweighing problem with the states from the previous ADMM iteration.

3.4.1 Weighted TV ADMM formulation

We begin by specifically solving Eq. (3.17) using ADMM at one particular iteration, i.e., with fixed γ_i's. The ADMM splitting technique is applied to decouple each convolved pixel, each isotropic TV element and each positive signal element, such that the optimization may be written as:

$$\begin{aligned} \underset{\boldsymbol{x}}{\arg\min} \;\; & \frac{1}{2}\left\|\boldsymbol{y}-\boldsymbol{M}\boldsymbol{u}\right\|_2^2 + \gamma_0 \sum_{i=1}^{N} \left(\gamma_i \|\boldsymbol{z}_i\|_2\right) + I_+(\boldsymbol{p}) \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x}=\boldsymbol{u}, \quad \boldsymbol{T}_i\boldsymbol{x}=\boldsymbol{z}_i, \quad \boldsymbol{x}=\boldsymbol{p}, \end{aligned} \tag{3.19}$$

where auxiliary variables $\boldsymbol{u} \in \mathbb{R}^N, \boldsymbol{z}_i \in \mathbb{R}^2, \boldsymbol{p} \in \mathbb{R}^N$ are introduced to split the problem into manageable parts, and $I_+(\cdot)$ is 0 if its argument is non-negative and ∞ otherwise.

The objective may be reformulated as the following *augmented Lagrangian*:

$$\begin{aligned}\mathcal{L}_{\rho_u,\rho_z,\rho_p}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{z},\boldsymbol{p},\boldsymbol{\eta},\boldsymbol{\lambda},\boldsymbol{\nu}) \overset{\text{def.}}{=} &\Big(\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{M}\boldsymbol{u}\|_2^2+\boldsymbol{\eta}^\top(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{u})+\frac{\rho_u}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{u}\|_2^2\Big)\\ &+\gamma_0\sum_{i=1}^{N}\Big(\gamma_i\|\boldsymbol{z}_i\|_2+\boldsymbol{\lambda}_i^\top(\boldsymbol{T}_i\boldsymbol{x}-\boldsymbol{z}_i)+\frac{\rho_z}{2}\|\boldsymbol{T}_i\boldsymbol{x}-\boldsymbol{z}_i\|_2^2\Big)\\ &+\Big(I_+(\boldsymbol{p})+\boldsymbol{\nu}^\top(\boldsymbol{x}-\boldsymbol{p})+\frac{\rho_p}{2}\|\boldsymbol{x}-\boldsymbol{p}\|_2^2\Big),\end{aligned} \tag{3.20}$$

where the dual variables are $\boldsymbol{\eta}\in\mathbb{R}^N$, $\boldsymbol{\lambda}_i\in\mathbb{R}^2$ and $\boldsymbol{\nu}\in\mathbb{R}^N$, and $\rho_u,\rho_z,\rho_p>0$ are the augmented parameters. The minimization of the augmented Lagrangian is approximately equal to (3.19) when $\rho_u,\rho_z,\rho_p\to\infty$. The first set of terms with $\boldsymbol{u},\boldsymbol{\eta}$ captures the splitting of the convolution from the mask. The second set of terms with $\boldsymbol{z},\boldsymbol{\lambda}$ captures the element-wise splitting of each edge term. The third set of terms with $\boldsymbol{p},\boldsymbol{\nu}$ captures the element-wise splitting of the non-negativity constraint. ADMM solves this iteratively by minimizing $\mathcal{L}_{\rho_u,\rho_z,\rho_p}$, at each step indexed by (t), first for $\boldsymbol{x}$, next for the groups $\{\boldsymbol{u},\boldsymbol{\eta}\}$,$\{\boldsymbol{z},\boldsymbol{\lambda}\}$, and $\{\boldsymbol{\nu},\boldsymbol{p}\}$ in parallel. Assumming that the unaugmented Lagrangian of (3.20) has a saddle point, and given that our objective function is closed, proper and convex, this weighted TV ADMM algorithm will attain convergence as $t\to\infty$ [22].

Minimization of $\boldsymbol{x}$

The minimization of $\boldsymbol{x}$ is reduced to a quadratic program with the closed form solution given by a Toeplitz matrix inversion

$$\begin{aligned}\boldsymbol{x}^{(t+1)} &\leftarrow \arg\min_{\boldsymbol{x}}\ \mathcal{L}(\boldsymbol{x},\boldsymbol{u}^{(t)},\boldsymbol{z}^{(t)},\boldsymbol{p}^{(t)},\boldsymbol{\eta}^{(t)},\boldsymbol{\lambda}^{(t)},\boldsymbol{\nu}^{(t)})\\ &=(\rho_u\boldsymbol{A}^\top\boldsymbol{A}+\gamma_0\rho_z\boldsymbol{T}^\top\boldsymbol{T}+\rho_p\boldsymbol{I})^{-1}\Big(\boldsymbol{A}^\top(\rho_u\boldsymbol{u}^{(t)}-\boldsymbol{\eta}^{(t)})\\ &\quad+\gamma_0\boldsymbol{T}^\top(\rho_z\boldsymbol{z}^{(t)}-\boldsymbol{\lambda}^{(t)})+(\rho_p\boldsymbol{p}^{(t)}-\boldsymbol{\nu}^{(t)})\Big),\end{aligned} \tag{3.21}$$

where $\boldsymbol{T}\in\mathbb{R}^{2N\times N}$ is a concatenation of all the $\boldsymbol{T}_i$'s. The Toeplitz structure of the inverse term permits an efficient solution by performing the inversion element-wise in the

Fourier space using the 2D fast Fourier transform (FFT). If non-periodic boundaries pose an issue, the 2D FFTs are applied with appropriate zero-padding. This is on the order of $O(N^{3/2} \log N)$ since it requires only one forward and one inverse FFTs (assuming the inverse matrix's FFT was pre-computed and cached).

Minimization of $\boldsymbol{u}$

The minimization of $\boldsymbol{u}$ is reduced to a quadratic program with a closed form solution that is solved in linear time

$$\begin{aligned} \boldsymbol{u}^{(t+1)} &\leftarrow \arg\min_{\boldsymbol{u}} \ \mathcal{L}(\boldsymbol{x}^{(t+1)}, \boldsymbol{u}, \boldsymbol{z}^{(t)}, \boldsymbol{p}^{(t)}, \boldsymbol{\eta}^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\nu}^{(t)}) \\ &= (\rho_u \boldsymbol{I} + \boldsymbol{M}^\top \boldsymbol{M})^{-1} (\boldsymbol{M}^\top \boldsymbol{y} + \rho_u (\boldsymbol{A}\boldsymbol{x}^{(t+1)} + \boldsymbol{\eta}^{(t)})). \end{aligned} \tag{3.22}$$

The inverse is element-wise separable since $\boldsymbol{M}^\top \boldsymbol{M}$ is a diagonal with 1's on the diagonal corresponding to the masking support and 0's corresponding to the off-support, hence it may be computed in $O(N)$.

Minimization of $\boldsymbol{z}$

The minimization of $\boldsymbol{z}$ is element-wise separable, and reduced to a proximal algorithm that is solved in linear time:

$$\begin{aligned} \boldsymbol{z}_i^{(t+1)} &\leftarrow \arg\min_{\boldsymbol{z}_i} \ \mathcal{L}(\boldsymbol{x}^{(t+1)}, \boldsymbol{u}^{(t)}, \boldsymbol{z}_i, \boldsymbol{p}^{(t)}, \boldsymbol{\eta}^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\nu}^{(t)}) \\ &= \arg\min_{\boldsymbol{z}_i} \ \left(\|\boldsymbol{z}_i\|_2 - \frac{\rho_z/\gamma_i}{2} \|\boldsymbol{z}_i - \left(\boldsymbol{T}_i \boldsymbol{x}^{(t+1)} + (1/\rho_z)\boldsymbol{\lambda}_i^{(t)}\right)\|_2^2 \right) \\ &= \mathrm{shrink}_{\rho_z/\gamma_i} \left(\boldsymbol{T}_i \boldsymbol{x}^{(t+1)} + (1/\rho_z)\boldsymbol{\lambda}_i^{(t)}\right), \end{aligned} \tag{3.23}$$

where the 2D shrinkage operator [132] (also known as the vector-soft function) is given by

$$\mathrm{shrink}_\rho(\boldsymbol{q}) = \max\left\{ \|\boldsymbol{q}\|_2 - \frac{1}{\rho}, 0 \right\} \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|_2}, \tag{3.24}$$

with the convention $\mathbf{0}/\|\mathbf{0}\|_2 = 0$.

Minimization of $\boldsymbol{p}$

The minimization of $\boldsymbol{p}$ is reduced to a proximal algorithm whose solution is the projection unto the non-negative orthant:

$$\begin{aligned}\boldsymbol{p}^{(t+1)} &\leftarrow \arg\min_{\boldsymbol{p}} \ \mathcal{L}(\boldsymbol{x}^{(t+1)}, \boldsymbol{u}^{(t)}, \boldsymbol{z}^{(t)}, \boldsymbol{p}, \boldsymbol{\eta}^{(t)}, \boldsymbol{\lambda}^{(t)}, \boldsymbol{\nu}^{(t)}) \\ &= \arg\min_{\boldsymbol{p}} \Big(I_+(\boldsymbol{p}) + (\rho_p/2)\|\boldsymbol{p} - (\boldsymbol{x}^{(t+1)} - (1/\rho_p)\boldsymbol{\nu}^{(t)})\|_2 \Big) \\ &= \max\{\boldsymbol{x}^{(t+1)} - (1/\rho_p)\boldsymbol{\nu}^{(t)}, \mathbf{0}\},\end{aligned} \tag{3.25}$$

that is also efficiently solved in linear time.

Update of dual variables $\boldsymbol{\eta}, \boldsymbol{\lambda}, \boldsymbol{\nu}$

The dual updates are updated with the following equations:

$$\boldsymbol{\eta}^{(t+1)} \leftarrow \boldsymbol{\eta}^{(t)} + \rho_u(\boldsymbol{A}\boldsymbol{x}^{(t+1)} - \boldsymbol{u}^{(t+1)}), \tag{3.26}$$

$$\boldsymbol{\lambda}_i^{(t+1)} \leftarrow \boldsymbol{\lambda}_i^{(t)} + \rho_z(\boldsymbol{T}_i\boldsymbol{x}^{(t+1)} - \boldsymbol{z}_i^{(t+1)}), \tag{3.27}$$

$$\boldsymbol{\nu}^{(t+1)} \leftarrow \boldsymbol{\nu}^{(t)} + \rho_p(\boldsymbol{x}^{(t+1)} - \boldsymbol{p}^{(t+1)}). \tag{3.28}$$

Over-relaxation

We applied the over-relaxation (OR) technique that was suggested by [22, 28] to accelerate ADMM convergence when the over-relaxation parameter $\alpha \in [1, 2)$ is properly tuned. In the updates given by Eq. (3.22) and (3.26), the quantity $\boldsymbol{A}\boldsymbol{x}^{(t+1)}$ is replaced by

$$\widehat{(\boldsymbol{A}\boldsymbol{x}^{(t+1)})} = \alpha\boldsymbol{A}\boldsymbol{x}^{(t+1)} - (1-\alpha)\boldsymbol{x}^{(t)}. \tag{3.29}$$

In the updates given by Eq. (3.23) and (3.27), the quantity $\boldsymbol{T}_i\boldsymbol{x}^{(t+1)}$ is replaced with

$$\widehat{(\boldsymbol{T}_i\boldsymbol{x}^{(t+1)})} = \alpha\boldsymbol{T}_i\boldsymbol{x}^{(t+1)} - (1-\alpha)\boldsymbol{z}_i^{(t)}. \tag{3.30}$$

In the $\boldsymbol{p}$ and $\boldsymbol{\nu}$ updates given by Eq. (3.25) and (3.28), the quantity $\boldsymbol{x}^{(t+1)}$ is replaced with

$$\widehat{(\boldsymbol{x}^{(t+1)})} = \alpha\boldsymbol{x}^{(t+1)} - (1-\alpha)\boldsymbol{p}^{(t)}. \tag{3.31}$$

3.4.2 Fused reweighted TV ADMM algorithm

In Algorithm 2, we propose performing reweighing at every T ADMM iterations, where T is much smaller than the number of ADMM iterations it would take for convergence of weighted TV to high accuracy. Although ADMM is slow to converge to high accuracy [136], it generally exhibits fast convergence to modest accuracy [22]. The idea is that a partially converged solution from few ADMM steps provides sufficient accuracy to produce reasonable weights $\boldsymbol{\gamma}$ for RWTV convergence; this will be empirically validated in section 3.4.4. Another advantage of this scheme is that it allows each subsequent ADMM inner-loop to warm-start using variables from the previous reweight. Finally, we impose a convergence stopping criteria of $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|_2 / \|\boldsymbol{x}^{(k)}\|_2 < \varepsilon$, where $\boldsymbol{x}^{(k)}$ refers to the $\boldsymbol{x}$ found at the k^{th} reweight, or if k reaches some user-defined K (i.e., maximum number of iterations), which ever comes first.

3.4.3 GPU implementation

The proposed RWTV-ADMM algorithm is distributed by design and can be accelerated by a graphics processing unit (GPU) implementation, whose numerous cores computes each of the distributed computations. For this work, we used the CUDA 8.0 Toolkit, which is a computing platform by NVIDIA for implementing parallel software on a GPU. One major bottleneck towards speed in GPU programming, is data-transfer overhead between the *host*

Algorithm 2 RWTV-ADMM

1: $\boldsymbol{x} \leftarrow \boldsymbol{u} \leftarrow \boldsymbol{z} \leftarrow \boldsymbol{p} \leftarrow \boldsymbol{\eta} \leftarrow \boldsymbol{\lambda} \leftarrow \boldsymbol{\nu} \leftarrow \mathbf{0}$
2: $k \leftarrow 0$
3: **repeat**
4: $\quad k \leftarrow k + 1$
5: $\quad t \leftarrow 0$
6: $\quad$ **repeat**
7: $\quad\quad t \leftarrow t + 1$
8: $\quad\quad$ update $\boldsymbol{x}$ by 3.21
9: $\quad\quad$ compute $(\widehat{\boldsymbol{A}\boldsymbol{x}^{(t+1)}})$ by (3.29)
10: $\quad\quad$ update $\boldsymbol{u}$ by (3.22) then $\boldsymbol{\eta}$ by (3.26) with OR
11: $\quad\quad$ compute $(\widehat{\boldsymbol{T}_i\boldsymbol{x}^{(t+1)}})$ by (3.30)
12: $\quad\quad$ update $\boldsymbol{z}_i$ by (3.23) then $\boldsymbol{\lambda}_i$ by (3.27) with OR $\forall i$
13: $\quad\quad$ compute $(\widehat{\boldsymbol{x}^{(t+1)}})$ by (3.31)
14: $\quad\quad$ update $\boldsymbol{p}$ by (3.25) then $\boldsymbol{\nu}$ by (3.28) with OR
15: $\quad$ **until** $t = T$
16: $\quad$ Update weights γ_i, $\forall i = 1, ..., N$ by 3.18
17: **until** $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|_2 / \|\boldsymbol{x}^{(k)}\|_2 < \varepsilon$ or $k = K$

(i.e., CPU) and the *device* (i.e., GPU). To reduce this, we only transfer data at the beginning and at the end of the algorithm. To push the envelope on computational speed, the algorithm was performed with floating point precision (instead of double precision) since NVIDIA's GPUs are accelerated at this precision (e.g., for graphics rendering).

A naïve implementation of matrix operations using full-sized matrices is memory-inefficient, especially since matrices in our application are highly structured. Instead, we encode matrices by their implicit operators. $\boldsymbol{A}$ being a convolution operator, may be implemented by direct convolution if the convolution kernel is small (e.g., 5×5) and this is efficiently parallelized on the GPU. $\boldsymbol{T}_i$ has ≤ 4 non-zero elements per row, so we exploit its structured sparsity and hardcode its operation.

Each ADMM step of Algorithm 2 (i.e., lines 7-14) may be decomposed into 3 stages. In the first stage, the signal $\boldsymbol{x}$ is updated (line 8) using CUDA's cuFFT library, which executes FFT at floating point efficiently [137]. In the second stage, the updates for the three set of variable pairs $\{\boldsymbol{u}, \boldsymbol{\eta}\}$ (lines 9-10), $\{\boldsymbol{z}, \boldsymbol{\lambda}\}$ (lines 11-12), and $\{\boldsymbol{p}, \boldsymbol{\nu}\}$ (lines 13-

14) are performed in parallel, using a single *kernel*[3], since terms do not interact across pairs. Auxiliary variables are updated followed by dual variables due to their serial dependence, with all variables updated element-wise for efficiency. In the final stage, the weights are updated element-wise (in line 16) using a single kernel. For the stopping criterion (line 17), we used CUDA's cuBLAS library for efficient ℓ_2 norm computation.

3.4.4 Numerical experiments

In this section, we apply the proposed RWTV-ADMM algorithm on a deconvolution and inpainting problem in differential interference contrast (DIC) microscopy for neuronal-cell segmentation [89]. Here, computer vision is required to assist in automated patch clamping [87] by providing precise cell-membrane localization (i.e., segmentation) in real-time (i.e., 5-10 frames per second). Specifically, this involves a deconvolution (using an assumed microscopy optics point spread function) of a 64×64 noisy DIC microscopy image patch containing a cell, followed by segmentation (using a global threshold obtained via Otsu's method [122]). We also require inpainting over a mask of the micro-pipette which would otherwise cause severe distortions on the reconstructed boundary. For data, we utilized 100 image trials generated from a previously developed cell simulator [91] (freely available under GNU GPL at http://siplab.gatech.edu) that produces realistic synthetic DIC cell imagery distorted by sensor noise (modeled as Gaussian and Laplacian distributions), along with ground truth.

In all trials, the ADMM parameters were tuned to $\rho_x = 10^{-5}, \rho_z = 10, \rho_p = 1, \alpha = 1.8$, the TV regularization parameter was set to $\gamma_0 = 3 \times 10^{-4}$, and convergence tolerance was set to $\varepsilon = 10^{-4}$ with a maximum of $K = 500$ reweights. Fig. 3.18 shows a sample of deconvolved images (with inpainting) from the dataset at various T. Qualitatively, the deconvolved images were visually similar in its segmentation boundary but the level of detail within the cell depended on T, where a lower T produced lower fidelity.

[3]In CUDA terminology, parallel instructions are distributed as *threads* to its multiple cores via a CUDA entity called the *kernel*.

Fig. 3.19 has plots of aggregated convergence and accuracy for various reweighing frequencies T from 100 image trials. Trials were performed on a MATLAB CPU implementation of RWTV-ADMM (at double precision). We used $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|_2 / \|\boldsymbol{x}^{(k)}\|_2$ to represent convergence, where $\boldsymbol{x}^{(k)}$ refers to the reconstruction found at the k^{th} reweight. Accuracy was represented by the relative mean square error (rMSE) metric, computed as $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}\|_2^2 / \|\boldsymbol{x}\|_2^2$, where $\boldsymbol{x}$ refers to the ground truth. To aid in visualization, we normalized durations for each trial into the range $[0, 1]$ via interpolation, since the number of iterations for each trial varied. Each line represents a particular T's average from 100 trials, while the error bars represent its ± 1 standard deviation. We observed that the proposed partial reweighing strategy converged at all values of T, even for $T = 1$. The accuracy of the converged solution was observed to be proportional to T, validating that premature reweighting (due to fewer ADMM iterations) still sacrifices accuracy. For T values higher than 8, negligible accuracy improvements were observed, suggesting that it is inefficient to converge to high accuracy before reweighing.

In Fig. 3.20, trials were performed using a CPU MATLAB implementation at double precision with an i7-4710HQ (4 cores clocked at 2.50 GHz), and using a GPU CUDA implementation at single precision with a GTX850 (640 parallel cores clocked at 936 MHz). Trials were repeated for both deconvolution only (i.e., $\boldsymbol{M} \equiv \boldsymbol{I}$), and deconvolution with inpainting. To validate the quality of our RWTV-ADMM optimization, we also ran a modification of the algorithm called RWTV-CVX (indicated in Fig. 3.20 by 'CVX'), where each inner-loop's weighted TV optimization is solved to high accuracy using an established third party solver, CVXOPT [128, 129] (implemented on CPU). For the RWTV-ADMM CPU implementation, we found a trade-off between runtime and accuracy, where runtime decreased with T while accuracy had the inverse relationship. Remarkably, at $T = 8$, accuracy of RWTV-ADMM was equivalent to that of RWTV-CVX, with a runtime improvement of 2 orders of magnitude. The maximum boundary error (MBE) metric defined in [89], is an application-specific metric that measures the maximum pixel-distance error around the

estimated contour segment, to highlight small boundary errors that might be harmful in the patch clamp application. All values of T produced MBEs better than RWTV-CVX, indicating that fast, precision membrane identification is attainable with RWTV-ADMM. For the GPU implementation, we observed that at $T = 8$, we were able to achieve the best runtime with up to 3 orders of magnitude of runtime improvement over RWTV-CVX, and attaining a runtime of 8-10 frames per second, along with better performance in both rMSE and MBE. On the downside, runtime increases were observed for $T = \{1, 2\}$ against $T = \{4, 8\}$, reflecting convergence issues that was attributed to the lowered single numerical precision.

3.5 Integration in automated patch-clamp system

For the final section in this chapter, we describe the integration of our membrane tracking algorithm into a larger robotic *PatcherBot* system. The PatcherBot system without feedback is error-prone due to the accumulation of mechanical errors in patch-clamp actuators, as well as cell motion primarily caused by tissue deformation during pipette entry. Machine vision (MV) feedback is therefore a critical piece of technology that corrects for these errors and cell motion in real time to improve throughput. Specifically, two components of machine vision were incorporated: (i) pipette auto-focus and cell auto-focus routines (via gross tracking algorithms such as template matching), and (ii) real-time membrane tracking (as developed in the previous sections) to compensate for minute displacement during the final approach of the pipette. In the final results, both components were subsequently benchmarked together.

The software use case is described as follows. A cell is first selected by the user and it's template is saved. It's DIC angle is estimated by the user and input as an algorithmic parameter. The cell template is then used to perform gross tracking (via template matching, i.e., cross correlation) to produce an image patch (approximately 64×64) containing the cell. The image patch is then processed with our algorithm to yield a segmented region,

for which its centroid is provided to the actuator system. While *PatcherBot* was written in LabVIEW, National Instruments, our software (written in MATLAB) interfaced it and had a refresh rate of 2-5 Hz, which was sufficient for real-time update feedback to the actuators.

Fig. 3.21(a), we show snapshots from a sample cell tracking video. More videos can be found in the supplementary materials of [95]. In Fig. 3.21(b)'s top plot, MV was found to successfully compensate for the motion in 133 cells out of 161 attempts (83%) where a successful compensation was defined as the algorithm reaching the *establish seal* state with the target cell. The *establish seal* state is defined as the moment when suction resistance between the pipette and the cell crosses some predefined threshold.

3.6 Discussion

We present the first deconvolution algorithm to locate cell boundaries with high precision in DIC microscopy images of brain slices. In summary, the main technical contributions of this algorithm are: (i) a pre-filtering step that is a computationally cheap and effective way at removing heavy organic interference with spectral characteristics, (ii) a dynamical ℓ_1 reweighing approach for the propagation of second-order edge statistics in online DIC cell segmentation, and (iii) an inpainting approach for pipette removal that is possible with little modification due to the inherently flexible framework of the algorithm. To quantitatively validate the performance of segmentation algorithms, we also describe a novel adaptation of cell simulation techniques to the specific data statistics of DIC microscopy imagery of brain slices in a publicly available MATLAB toolbox.

The proposed algorithm achieves state-of-the-art performance in tracking the boundary locations of cells, with the average and maximum boundary errors achieving the desired tolerances (i.e., 1-2 μm) driven by the accuracy of actuators used for automatic pipette movement. These results lead us to conclude that accurate visual guidance will be possible for automated patch clamp systems, resulting in a significant step toward high-throughput brain slice electrophysiology. The main shortfall of the proposed algorithm arises from the

implicit assumption that dynamics are spatially limited. In order for dynamics to positively contribute to the deconvolution process, edges in the current frame should fall within the vicinity of the previous frame's edges; specifically, it should be bounded by the support of the prediction kernel as described by (3.15). Future work aims at investigating alternative methods that incorporate dynamics in a fashion that is not spatially-limiting, and characterizing dynamical functions that take into account physical models of motion in the system (e.g., motion induced by the pipette, fluid dynamics, cell deformation physics).

To realize our algorithm, we developed a general efficient ADMM solver framework for the reweighted total variation (RWTV) algorithm. We demonstrated that the RWTV algorithm could be significantly accelerated by performing reweights only after partial convergence, and by warm-starting the ADMM algorithm with state variables from the previous reweight. Since the proposed ADMM algorithm was also parallelizable, we implemented a GPU-accelerated version of the algorithm and demonstrated significant speed increases (from a CPU-version, and from CVX) that were necessary for a real-time computer vision application in microscopy cell-tracking. Though the presented framework was cast in a reweighted TV minimization setting here, it could be easily extended to more general settings such as reweighted ℓ_1 or reweighted nuclear-norm.

Finally, we packaged our algorithm as a software and interfaced it with *PatcherBot* so that automated patch clamping trials could be conducted at the level of an integrated robotic system. On the overall, we demonstrated that computer vision significantly increased yield by actively compensating for XY displacement errors (even on the order of $2\mu m$). While the current work is promising, the XY-only microscope imaging plane neglects the Z-axis, which causes a host of issues. For one, the tracking procedure is sensitive to out-of-focus scenarios, which is easily caused by pipette motion. Another issue is that the mismatch between the pipette tip and the imaging plane renders precise boundary information unimportant (centroid information is often found to be better than boundary information due to this mismatch). To remedy these issues, the imaging setup should, for future work, be

extended to the multi-plane setting in XYZ. This would not only alleviate out-of-focus issues, it would also truly exploit precise boundary information. Moreover, this would in principle tell us the optimal "angle of attack" of the pipette's approach. For this advance to occur, however, new computational strategies are needed to ease the increased computational burden. One strategy, for example, could be an online decentralized implementation that processes the multiple planes in the Z-axis somewhat independently, only *reducing* their information at infrequent intervals to reduce communication cost.

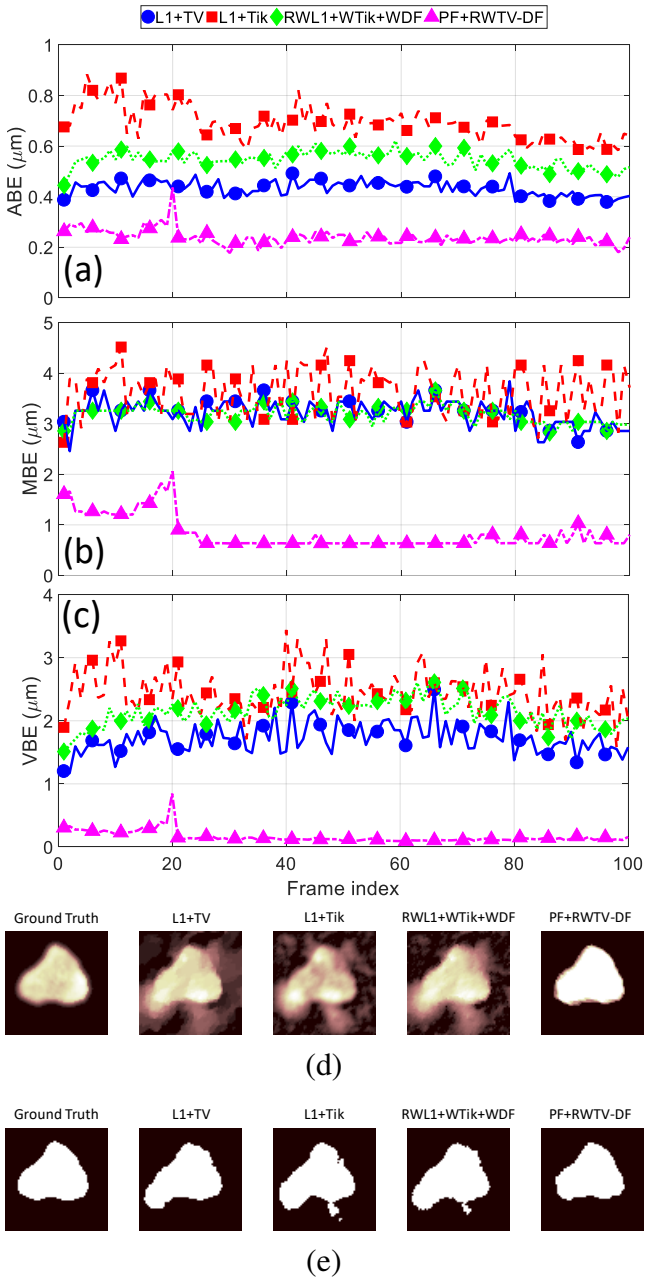


Figure 3.15: ABE, MBE, and VBE evolving with time are shown from (a) to (c) (respectively) for a single representative video (Trial #4) of a simulated DIC microscopy cell with 100 video frames after deconvolution and segmentation using four different algorithms. The images in (d) are snapshots of the deconvolution output at frame 100, and the images in (e) are their respective segmentations.

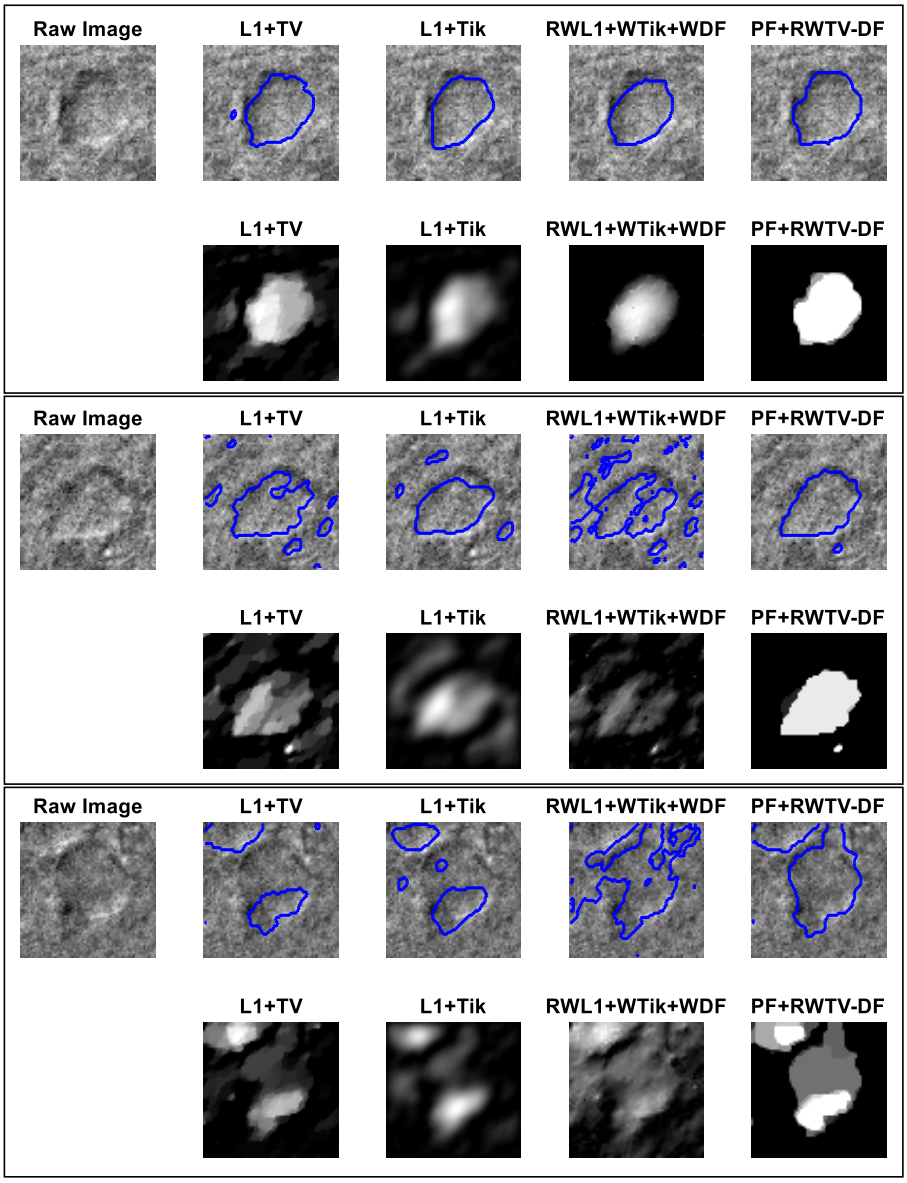


Figure 3.16: Three example image patches of real cell data were deconvolved using four algorithms and the resulting segmentations (top row) and deconvolutions (row below) are displayed. Segmentations were obtained from deconvolution by global thresholding using Otsu's method [122]. The proposed PF+RWTV-DF algorithm performs consistently well even in severe interference (defined by the degree of distortion around the edges and surface undulations within the cell).

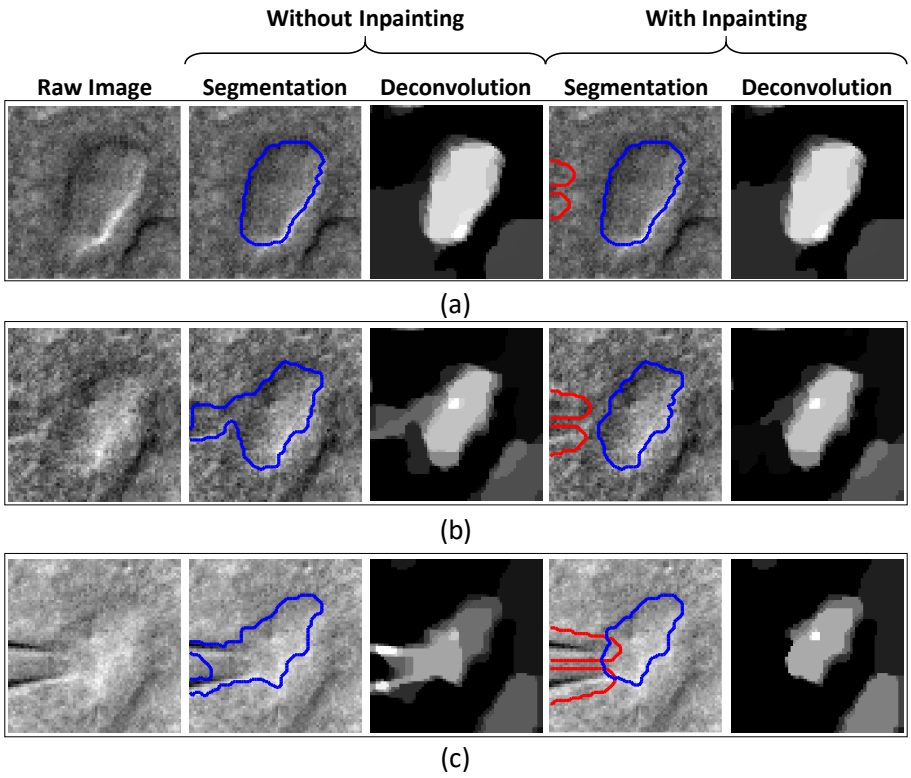


Figure 3.17: Snapshots from real video data of a cell undergoing patch clamping demonstrate the efficacy of the proposed pipette removal method (via inpainting). The pipette mask is outlined in red while the cell segmentation is outlined in blue. Each snapshot illustrated varying degrees of overlap between the pipette and the cell: (a) the pipette is far from the cell, (b) the pipette is in close proximity to the cell, and (c) the pipette is "touching" the cell. The efficacy of the proposed inpainting method for pipette removal (versus no inpainting) is qualitatively demonstrated by its ability to cleanly segment the cell despite the pipette's presence.

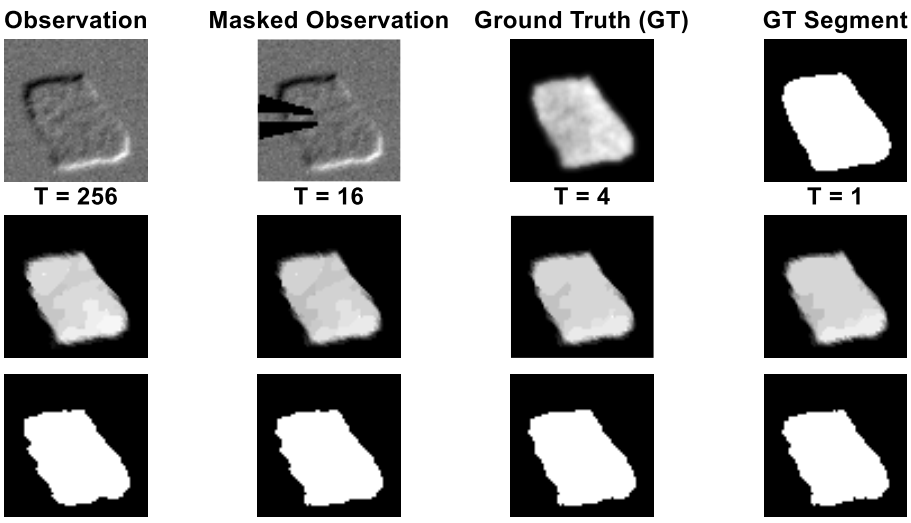


Figure 3.18: Deconvolution with inpainting for a single trial at four T values. First row: observation without and with mask (i.e., elongated black triangles indicating masking of pipette) and the ground truth alongside its binary segmentation. Second row: reconstructions at various reweighing frequencies T with its respective segmentation below.

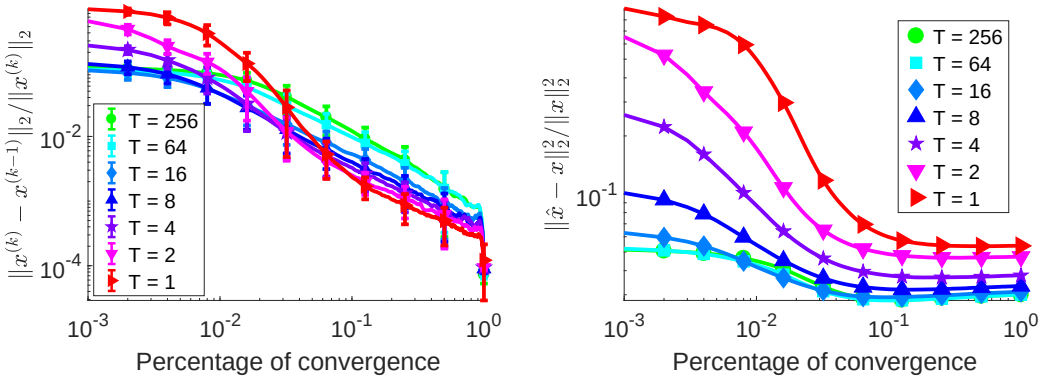


Figure 3.19: Aggregated convergence and accuracy plots for various reweighing frequencies (T) from 100 trials of deconvolution with inpainting. Left: duration-normalized convergence plots. Right: duration-normalized relative mean square error (rMSE) plots.

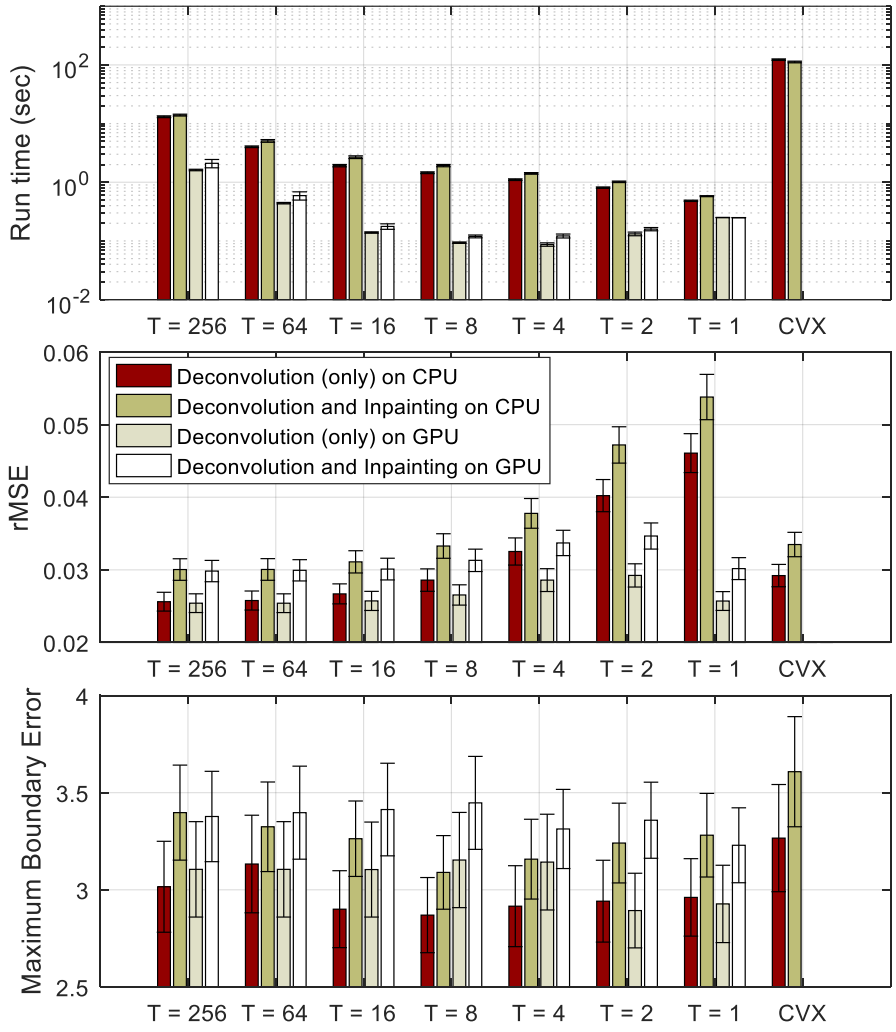
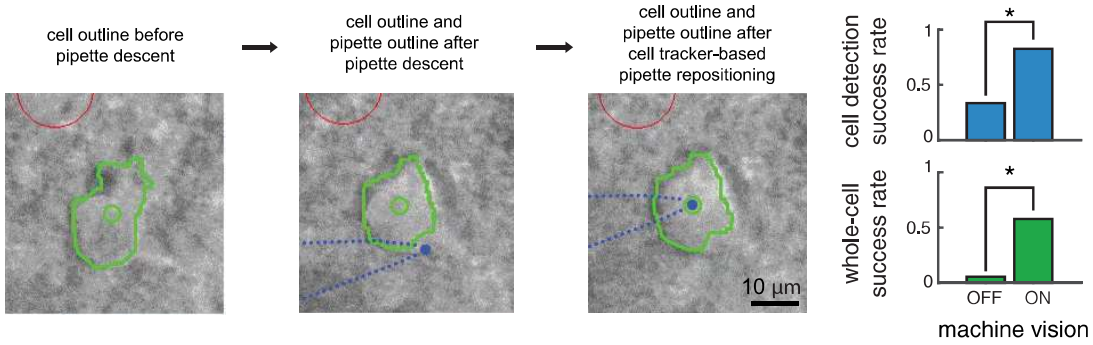


Figure 3.20: Aggregated statistics of RWTV-ADMM across various T for 100 trials.Top: Runtime for a single image solve. Middle: Accuracy measured by the relative mean square error. Bottom: Segmentation accuracy measured by the maximum boundary error.



(a) (b)

Figure 3.21: Cell tracking results in patch clamp system: (a) sample cell tracking results, (b) experimental yield. In (a), cell boundaries (green outline) are automatically tracked and the centroid (green circle) is computed. Pipette tip location is estimated from the manipulator position (blue dot). Left: cell boundary before pipette descent. Center: cell position and pipette position after pipette descent. Right: cell position and pipette position after trajectory adjustment. In (b), we report the success rate of cell detection and whole-cell recordings with and without machine vision (off: $n = 18$ attempts, on: $n = 161$ attempts). P-values for fisher's exact test: cell detection $P = 2.4 \times 10^{-5}$, whole-cell $P = 1.7 \times 10^{-5}$.

CHAPTER 4

SCALABLE 1-WASSERSTEIN REGULARIZATION FOR INVERSE PROBLEMS

While there has been much progress in inverse imaging, fueled by advances in numerical optimization [134, 138, 24] and theoretical recovery guarantees [3, 4, 139], the modelling of physical phenomena remains a crucial aspect in solving such problems. Many inverse imaging problems (e.g., denoising, deconvolution, inpainting) are cast as optimization programs and are solved by exploiting *a priori* structure through the construction of meaningful modelling regularizers. Regularizers that employ ℓ_p metrics are often useful at modeling such phenomena because they efficiently describe point-wise statistics under fixed grid representations (e.g., rectangular pixel layout in imaging). However, ℓ_p metrics are sometimes inadequate because they fundamentally lack the mechanism to capture geometric relations between its support coordinates. In contrast, the optimal transport (OT) problem is a framework that explicitly accounts for geometric relationships by modelling a signal as mass that incurs a cost to move around its support. Under certain geodesic metrics (e.g. Euclidean notions of geometry in the support), the OT framework (e.g., Wasserstein distance) very naturally quantifies uncertainty and deformation. As a result of these attractive properties, computational approaches to regularization with OT have found a variety of imaging applications such as incompressible fluid flow [140, 141], temporal dynamics of sparse signals [142, 143], and physical deformation in medical images [46].

Despite OT's favorable modelling capabilities, two major drawbacks exist. First, evaluating OT is traditionally computationally expensive, limiting the state-space size N (e.g., number of pixels) in practice. In the most general formulation, solving OT amounts to solving an LP with N^2 variables and even efficient solvers (e.g., interior-point or simplex methods) have a computational complexity of at least $O(N^3)$. The recent computational advances in OT are partially attributed to the introduction of *Sinkhorn distances*

[63] as an efficient approximation method that trades-off accuracy for efficiency, reducing the per iteration costs to $O(N^2)$ (see [43] and references therein). Unfortunately, this trade off can be potentially counterproductive in applications that demand both accuracy and speed. Under a more restrictive case where the underlying geodesic metric is assumed to be Euclidean, Beckmann's formulation [54] offers a significantly smaller optimization space than Sinkhorn methods ($O(N)$ versus $O(N^2)$) without sacrificing accuracy. The Euclidean geodesic restriction is arguably a reasonable modeling assumption in imaging and video, by virtue of the success of optical flow methods which hold similar assumptions. Recently, this brand of OT has been employed with remarkable efficiency for large-scale images [53, 144].

The second major drawback of the traditional OT formulation is a modelling deficiency: it restricts applications to the space of *balanced* (i.e., fixed) mass systems (e.g., histograms and incompressible fluids), due to the assumption of conservation of mass. When applied to images and video, this constraint is restrictive as intensity invariably changes (e.g., when non-spherical objects rotate or when luminescence varies). To tackle this limitation, one may employ strategies such as *partial* transport [55, 59], which transports only a limited amount of mass between the systems, or *unbalanced* transport [56, 57], which additionally models statistical properties of mass growth and decay to account for mass differences.

We simultaneously tackle these two specific drawbacks by (i) enriching OT's modelling capabilities with unnormalized constraints, and (ii) significantly reducing the optimization size from $O(N^2)$ to $O(N)$ to make it tractable for large-scale inverse imaging applications. In this chapter, our contributions[1] are

1. a Beckmann formulation of the partial transport model for the optimal transport reg-

[1] This work was performed in collaboration with Adam S. Charles, Nicholas P. Bertrand, Pavel B. Dunn, Christopher J. Rozell. For the work on partial transport regularization [142, 97, 145, 146], ASC, NPB, JL were responsible for the algorithmic formulation of BPDN-EMD, PBD was responsible for synthetic simulation generation, and JL was responsible for the development of fast Beckmann formulations. For the work on unbalanced transport regularization, JL was responsible for the algorithmic formulations, the UOT proximal primal-dual algorithm, the ADMM solver for RPCA+UOT, and simulations on synthetic data, while NPB was responsible for the convergence proof for the proximal algorithm, and simulations on real data. CJR supervised the project.

ularized BPDN problem;

2. a novel Beckmann formulation of the recent *unbalanced* transport model [56], which has more descriptive statistics than partial transport; and

3. a parallelizable, and provably convergent iterative proximal algorithm for the proposed unbalanced-OT Beckmann formulation, which facilitates compatibility with first-order solvers that are a modern staple in the image processing toolbox.

To illustrate the utility of our proposed approach, we demonstrate superior reconstruction and support recovery performance in two dynamical tracking inverse imaging applications: (i) an online pixel-tracking estimation algorithm similar to dynamical methods such as [147, 148], and (ii) a novel dynamical tracking method that augments robust PCA [139] with our unbalanced OT regularizer, along with a distributed ADMM solver for this problem. We note that concurrent (preliminary) work describing a similar unbalanced OT formulation in isolation – not as part of the general proximal framework we propose – was concurrently developed [149].

4.1 Dynamical tracking: problem definition and background

We showcase our optimal transport model as a regularizer in *dynamical tracking* applications [148]; to do so, we introduce novel tracking models that encourage mass transport *continuity* between snapshots of time-varying data. We represent snapshots of target activations with the state vector $\boldsymbol{s}_t \in \mathbb{R}^N$ where t is the time index. Targets are assumed mobile and K-sparse over a gridded support where $\boldsymbol{s}_t$ evolves dynamically in time and space via a first-order Markovian relationship through some function f_t

$$\boldsymbol{s}_t = f_t(\boldsymbol{s}_{t-1}) + \boldsymbol{\nu}_t,$$

where $\boldsymbol{\nu}_t \in \mathbb{R}^N$ captures dynamical innovations (i.e., model mismatch).

We consider the problem of dynamically tracking $\boldsymbol{s}_t$ from streaming measurements

$$\boldsymbol{y}_t = \boldsymbol{\Phi}_t \boldsymbol{s}_t + \boldsymbol{\eta}_t,$$

where $\boldsymbol{\eta}_t \in \mathbb{R}^M$ represents an iid Gaussian noise model. The goal in this tracking problem is to accurately recover $\boldsymbol{s}_t$ at each time step given $\boldsymbol{\Phi}_t$ and $\boldsymbol{y}_t$.

Classical dynamical tracking methods like the Kalman filter [6] exploit temporal structure for estimation and inference. Under a probabilistic *maximum a posteriori* framework, the Kalman filter provides the optimal[2] estimate of the current signal under a linear dynamical function (i.e., $f_t(\boldsymbol{s}_{t-1}) = \boldsymbol{F}_t \boldsymbol{s}_{t-1}$). Concisely stated, the Kalman filter solves

$$\widehat{\boldsymbol{s}}_t = \underset{\boldsymbol{s}}{\arg\min} \ \|\boldsymbol{y}_t - \boldsymbol{\Phi}_t \boldsymbol{s}\| + \|\boldsymbol{s} - \boldsymbol{F}_t \widehat{\boldsymbol{s}}_{t-1}\|,$$

which balances an observation term and a prediction term (using norms that capture noise statistics). The ℓ_p losses only make design-sense under a *Lagrangian* state-space representation where each state variable $\boldsymbol{s}_t[i]$ is itself moving through a geometric space (e.g., displacement coordinates of a GPS sensor). A design-flaw arises if the state space is *Eulerian*-represented (i.e., the support space is gridded and signals move across its support). When displacement-variations are expected, it makes sense to pay penalties that are proportional to support-displacement error, yet ℓ_p-norm penalties are invariant to this, rendering ℓ_p-norms ineffective at support estimation. In spite of this, *Eulerian*-represented signals are still of interest because they are extremely effective when signal-support statistics are known *a priori* (e.g., sparsity statistics [3]). The unbalanced-OT model is a suitable regularizer that circumvents support-estimation issues, since it inherently accounts for the geometry of an *Eulerian* representation's support. We also note that optimal transport priors have recently been proposed in inverse imaging problems [46, 45], though not in dynamical settings.

[2] Under Gaussian assumptions on the measurement and dynamical noise.

4.2 1-Wasserstein partial transport formulation

We recall the Monge-Kantorovich optimal transport formulation (from Chapter 2):

$$\mathcal{W}(\boldsymbol{p},\boldsymbol{q}) = \min_{\boldsymbol{P}\geq 0} \ \langle \boldsymbol{P},\boldsymbol{C}\rangle \quad \text{s.t.} \quad \boldsymbol{P}\mathbb{1} = \boldsymbol{p}, \boldsymbol{P}^\top\mathbb{1} = \boldsymbol{q}, \tag{OT}$$

where $\boldsymbol{p},\boldsymbol{q} \in \mathbb{R}^N$ are input marginals of the optimal transport program, $\boldsymbol{P} \in \mathbb{R}_+^{N\times N}$ is the transport coupling, and $\boldsymbol{C} \in \mathbb{R}^{N\times N}$ is the cost distance matrix. In this section, we develop a *partial* transport [55, 37] strategy for variational problems. Compared to (OT), partial transport limits its transportation budget to only a fraction of mass of its arguments via the additional constraint $\mathbb{1}^\top\boldsymbol{P}\mathbb{1} \leq \min(\mathbb{1}^\top\boldsymbol{p}, \mathbb{1}^\top\boldsymbol{q})$. To convexify this constraint with respect to $\boldsymbol{p}$ and $\boldsymbol{q}$, we propose a reformulation to control the transport budget variationally using a parameter $\mu > 0$ that regulates the transport budget ($\mathbb{1}^\top\boldsymbol{P}\mathbb{1}$):

$$\begin{aligned}\mathcal{U}_\mu(\boldsymbol{p},\boldsymbol{q}) = \min_{\boldsymbol{P}\geq 0} \ & \langle \boldsymbol{P},\boldsymbol{C}\rangle - \mu\mathbb{1}^\top\boldsymbol{P}\mathbb{1} \\ \text{s.t} \ & \boldsymbol{P}\mathbb{1} \leq \boldsymbol{p}, \ \ \boldsymbol{P}^\top\mathbb{1} \leq \boldsymbol{q}, \\ & \mathbb{1}^\top\boldsymbol{P}\mathbb{1} \leq \mathbb{1}^\top\boldsymbol{p}, \ \ \mathbb{1}^\top\boldsymbol{P}\mathbb{1} \leq \mathbb{1}^\top\boldsymbol{q}.\end{aligned} \tag{4.1}$$

In this formulation, the objective function's second term regulates the transport budget by *maximizing* the total budget, while additional constraints are included to ensure the total budget does not exceed the mass contained by the input marginals.

We extend Beckmann's formulation (2.5) with the partial transport constraints:

$$\begin{aligned}\widetilde{\mathcal{U}}_\mu(\boldsymbol{p},\boldsymbol{q}) = \min_{\boldsymbol{M},\widetilde{\boldsymbol{p}},\widetilde{\boldsymbol{q}},u} \ & \|\boldsymbol{M}\|_{2,1} - \mu u \\ \text{s.t} \ & \text{div}(\boldsymbol{M}) - \widetilde{\boldsymbol{p}} + \widetilde{\boldsymbol{q}} = \boldsymbol{0} \\ & \boldsymbol{0} \leq \widetilde{\boldsymbol{p}} \leq \boldsymbol{p}, \ \ \boldsymbol{0} \leq \widetilde{\boldsymbol{q}} \leq \boldsymbol{q}, \\ & \mathbb{1}^\top\widetilde{\boldsymbol{p}} = \mathbb{1}^\top\widetilde{\boldsymbol{q}} = u, \\ & u \leq \mathbb{1}^\top\boldsymbol{p}, \ \ u \leq \mathbb{1}^\top\boldsymbol{q},\end{aligned} \tag{4.2}$$

where $\boldsymbol{M} \in \mathbb{R}^{N\times d}$ represents a vector field, $\widetilde{\boldsymbol{p}}, \widetilde{\boldsymbol{q}} \in \mathbb{R}^n_+$ are nonnegative vectors with similar dimensions as $\boldsymbol{p}, \boldsymbol{q}$, and $u \in \mathbb{R}_+$ is the transport budget variable. This optimization searches for the minimal vector field configuration that describes, via the first constraint, its flux to be travelling between a source $\widetilde{\boldsymbol{p}}$ and a sink $\widetilde{\boldsymbol{v}}$. The second and third constraint describes the source and sink as nonnegative slack variables that are bounded above by their proxies $\boldsymbol{p}$ and $\boldsymbol{q}$ respectively; this constraint is analogous to the mass preservation constraints in (4.1). The last three constraints state that the induced flux must be bounded by the total transport budget, which is similar spirit in to the last two constraints of (4.1). Compared to the N^2 variables that (4.1) has, this formulation has only $N(d+2)$ variables, where d is the dimensions of the vector field (e.g., $d = 2$ for images). One downside of this formulation is sheer number of constraints, which makes it cumbersome to implement; we will address this issue with our unbalanced formulation in the next section.

4.2.1 Sparse tracking of signals in video

Incorporating *partial* OT into the dynamic filtering, and defining $\widetilde{\boldsymbol{x}}_t = f(\widehat{\boldsymbol{x}}_{t-1})$ as the predicted signal, yields an BPDN+EMD optimization formulated as

$$\begin{aligned}
\widehat{\boldsymbol{x}}_t &= \underset{\boldsymbol{x}}{\arg\min} \;\; \|\boldsymbol{y}_t - \boldsymbol{\Phi}_t\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 + \gamma\mathcal{U}_\mu(\boldsymbol{x}, \widetilde{\boldsymbol{x}}_t) \\
&= \underset{\boldsymbol{x},\boldsymbol{P}}{\arg\min} \;\; \|\boldsymbol{y}_t - \boldsymbol{\Phi}_t\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 + \gamma\langle \boldsymbol{P}, \boldsymbol{C}\rangle - \mu\mathbb{1}^\top\boldsymbol{P}\mathbb{1} \qquad (4.3) \\
&\qquad \text{s.t.} \qquad \boldsymbol{P}\mathbb{1} \le \boldsymbol{x}, \boldsymbol{P}^\top\mathbb{1} \le \widetilde{\boldsymbol{x}}_t, \mathbb{1}^\top\boldsymbol{P}\mathbb{1} \le \mathbb{1}^\top\boldsymbol{x}, \mathbb{1}^\top\boldsymbol{P}\mathbb{1} \le \mathbb{1}^\top\widetilde{\boldsymbol{x}}_t.
\end{aligned}$$

This cost function can be jointly optimized over both $\boldsymbol{x}$ and $\boldsymbol{P}$ to find the overall new, current signal estimate $\widehat{\boldsymbol{x}}_t$. We test our method by implementing our optimization program in CVX [128, 129], and explore the utility of EMD-regularized BPDN over other sparsity-based tracking algorithms, BPDN-DF [150, 15] and re-weighted ℓ_1 dynamic filtering (RWL1-DF) [15]. To maximize performance of all algorithms in the experiments, a geometric grid search was performed over parameters λ and γ to minimize the relative

mean-squared error (rMSE) given by $\|\widehat{\boldsymbol{x}} - \boldsymbol{x}\|_2^2 / \|\boldsymbol{x}\|_2^2$ where $\widehat{\boldsymbol{x}}$ is the estimation and $\boldsymbol{x}$ is the true signal.

We first applied a one-step recovery using compressive measurements on a relatively large simulation example (Figure 4.1) where K synthetic targets moved randomly within a 1-pixel neighborhood between frames. As the dynamics model was not well specified, the first frame's ground truth was used as the signal prediction. In this task, BPDN-EMD significantly outperforms BPDN-DF and RWL1-DF at exploiting dynamical information and recovering the target locations. When a full sequence of frames were run in succession (Figure 4.2), BPDN-EMD quickly converges to a lower rMSE than any of the other methods, and tracks the targets more accurately over the entire time-course. Finally, we present Donoho-Tanner phase transition diagrams (Figure 4.3) to illustrate algorithm performance across different numbers of measurements and signal sparsities. These diagrams contextualize the previous experiments (which was conducted at specific M, K), demonstrating that BPDN-EMD achieves superior performance over competing methods at lower measurement numbers and higher sparsities.

To conclude, we note that this work has been successfully applied for the frequency tracking of EEG time-series signals, for which more details can be found in [143, 146].

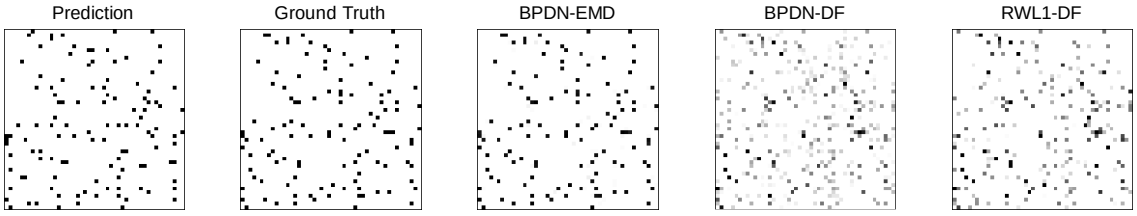


Figure 4.1: Example recovery of sparse, moving targets from compressive measurements. Recovering 48×48 pixel-sparse images demonstrates high accuracy of recovery from the EMD-regularized BPDN algorithm. Here, dynamic-aware algorithms are given a prediction signal (i.e., the previous frame from a dynamic sequence) to assist at inferring the ground truth signal, both of which are shown above. In this simulation scenario ($N = 2304, M = 0.15N, K = 0.33M$), the EMD-regularized BPDN recovers the signal most accurately.

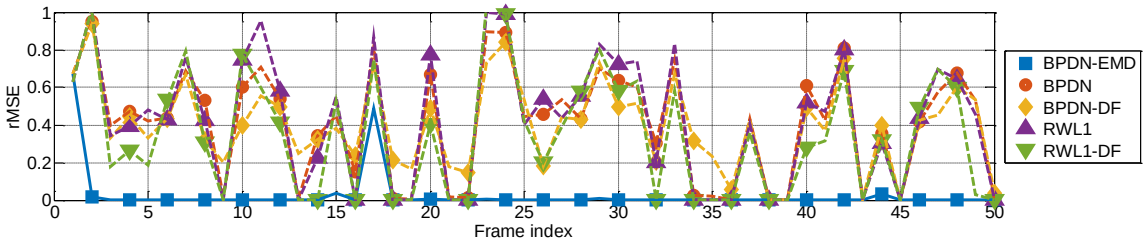


Figure 4.2: Recovery of a sequence of moving-targets, simulated with pseudo-Brownian motion where targets move randomly within a one-pixel neighborhood (between frames). Compressive measurements were used to recover the sequence using basis-pursuit denoising (BPDN), BPDN-DF, re-weighted ℓ_1 (RWL1) and RWL1-DF. Algorithmic accuracy was calculated with the mean-squared error. For dynamic-aware algorithms, the prediction signal is taken as the algorithm's previous estimate. In this simulation scenario ($N = 256, M = 0.15N, K = 0.33M$), the EMD-regularized BPDN achieves the lowest overall rMSE with remarkable robustness. Although the other dynamic filtering algorithms (BPDN-DF and RWL1-DF) demonstrated slight improvements over their non-dynamic versions (BPDN and RWL1), they did not have the robustness of BPDN-EMD.

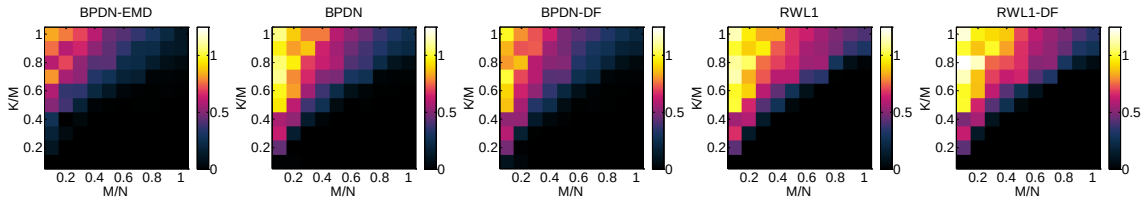


Figure 4.3: One-step recovery results as Donoho-Tanner phase transition diagram. Here, $N = 100$ and each point in each image was generated from the mean rMSE of 10 independent one-step recovery simulations. These diagrams illustrate EMD-regularized BPDN's superior rMSE performance in the space of M, K, as compared to other algorithms.

4.2.2 Computational Scalability

Given the increases in performance and robustness demonstrated by partial OT (4.1), we are especially interested in improving computational complexity so that the algorithm can still scale well in practical applications with large state spaces. Here we examine the impact of adapting an approach based on Beckmann's formulation (4.2) into our tracking problem.

Beckmann's formulation of BPDN-EMD is

$$\begin{aligned} \widehat{\boldsymbol{x}}_t = \underset{\boldsymbol{x},\boldsymbol{M},\boldsymbol{z},\widetilde{\boldsymbol{z}},u}{\arg\min} \quad & \|\boldsymbol{y}_t - \boldsymbol{\Phi}_t \boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 + \gamma \|\boldsymbol{M}\|_{2,1} - \mu u \\ \text{s.t.} \quad & \operatorname{div}(\boldsymbol{M}) - \boldsymbol{z} + \widetilde{\boldsymbol{z}} = \boldsymbol{0}, \\ & \boldsymbol{0} \leq \boldsymbol{z} \leq \boldsymbol{x}, \boldsymbol{0} \leq \widetilde{\boldsymbol{z}} \leq \widetilde{\boldsymbol{x}}_t, \\ & \mathbb{1}^\top \boldsymbol{z} = \mathbb{1}^\top \widetilde{\boldsymbol{z}} = u, \\ & u \leq \mathbb{1}^\top \boldsymbol{x}, u \leq \mathbb{1}^\top \widetilde{\boldsymbol{x}}_t. \end{aligned} \tag{4.4}$$

We conducted a similar simulation detailed in the previous section and scaled the problem between state sizes of 12×12 ($N = 144$) and 48×48 ($N = 2304$), where N is the total number of state elements. For each state size, the sparsity level was fixed at 5%. Each experiment was repeated 10 times for statistics aggregation and error bars denote ± 1 standard deviation from the mean. Because our major concern is whether the proposed computational modification degrades BPDN-EMD performance over the general (but expensive) formulation, we compare differences *between* solutions using the root mean squared error (RMSE): $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2}$, where x_i and y_i are the values of state elements in the respective solutions. We use the MOSEK solver for both formulations for a fair comparison and measure relative runtime on a personal computer (Intel Core i7 with 3.5 GHz processor speed).

Figure 4.4 shows that by using the Beckmann's formulation of BPDN-EMD (4.4), we obtain a significant speed up over the general formulation of BPDN-EMD (4.3). Figure 4.4 also shows that the difference between the Beckmann's formulation of BPDN-EMD and the general formulation have very small differences. Taken together, these results demonstrate that the proposed re-formulation is much more computationally tractable and scalable to larger problem sizes while producing solutions that are essentially the same as the general approach.

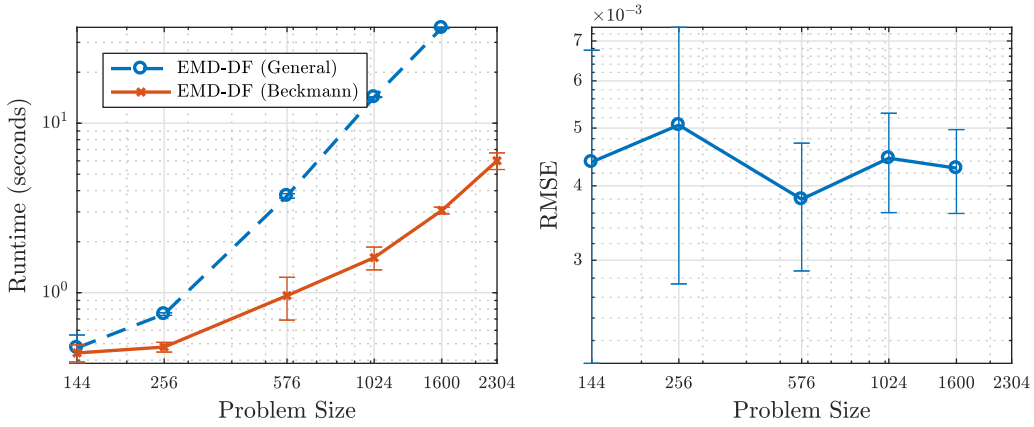


Figure 4.4: Demonstration of computational speed up with comparable solutions. We compare the runtime and the difference in solutions for two formulations of BPDN-EMD: BPDN-EMD (4.3) (labeled as EMD-DF (General)) which adopts generic distance costs, and the Beckmann version of it (4.4) (labeled as EMD-DF (Beckmann)) which assumes Euclidean distance costs. The left plot demonstrates that BPDN-EMD (Beckmann) significantly outperforms BPDN-EMD (General) in runtime, and in the right plot, the difference in solutions were shown to be negligible.

4.3 1-Wasserstein unbalanced transport formulation

Another recent strategy [56, 57] models statistical properties of unaccounted mass via a mechanism of growth and decay. We recall the *unbalanced* transport formulation (4.5) from Chapter 2:

$$\mathcal{V}_\mu(\boldsymbol{p}, \boldsymbol{q}) = \min_{\boldsymbol{P} \geq 0} \langle \boldsymbol{P}, \boldsymbol{C} \rangle + \mu(\|\boldsymbol{P}\mathbb{1} - \boldsymbol{p}\|_p^p + \|\boldsymbol{P}^\top \mathbb{1} - \boldsymbol{q}\|_p^p). \tag{4.5}$$

The terms weighed by parameter $\mu > 0$ penalize unaccounted mass between the marginals of the optimal transport coupling and the input arguments, which analogously regulates the transport budget. Although (4.1) and (4.5) both possess mechanism to regulate the transportation budget, (4.5)'s ℓ_p-norms explicitly model growth/decay statistics while (4.1) models only the gross growth/decay budget. For this reason, we advocate an unbalanced transport strategy, and we provide empirical evidence in Section 4.3.1 showing its advantage in an inverse imaging tracking problem.

We define the Beckmann formulation for *unbalanced* optimal transport as

$$\begin{aligned} \widetilde{\mathcal{V}}_\mu(\boldsymbol{p},\boldsymbol{q}) = \min_{\boldsymbol{M},\boldsymbol{r}} \ & \|\boldsymbol{M}\|_{2,1} + \mu\|\boldsymbol{r}\|_p^p \\ \text{s.t. } & \operatorname{div}(\boldsymbol{M}) - \boldsymbol{q} + \boldsymbol{p} = \boldsymbol{r}. \end{aligned} \tag{4.6}$$

The key difference between this formulation and the Beckmann OT formulation (2.5) is that there now exists a *transport residual* term $\boldsymbol{r} \in \mathbb{R}^N$ that reflects the amount of mass that needs to be created or destroyed (i.e., unaccounted flux divergence) to balance the equality constraint. To prevent unabated growth and decay of $\boldsymbol{r}$, we penalize its magnitude with an ℓ_p^p norm, where p is chosen based on statistics of the signal's support. For example, if $\boldsymbol{p}, \boldsymbol{q}$ are assumed sparse (i.e., $\|\boldsymbol{p}\|_0 \approx \|\boldsymbol{q}\|_0 \ll N$), we select $p = 1$ to reflect a kurtotic distribution over the transport residual's support, meaning that growth/decay can only occur at a sparse number of locations. Transportation cost is balanced against the cost of growth/decay using parameter $\mu > 0$. We remark that in the same way that (4.5) is a generalization of (OT), (4.6) is also a generalization of (2.5): a large μ drives the transport residual to a small value, therefore $\mathcal{V}_{\mu\to\infty} \to \mathcal{W}$. We also note that compared to the partial transport formulation (4.2), its single equality constraint makes it significantly easier to solve numerically (compared to multiple inequality constraints).

4.3.1 Experiments with OT-regularization under mass-changing regimes

We propose a least-squares dynamic filtering formulation with unbalanced OT-regularization (UOT-DF) under assumptions of first-order Markovian dynamics:

$$\widehat{\boldsymbol{s}}_t = \underset{\boldsymbol{s}\geq 0}{\arg\min} \ \frac{1}{2}\|\boldsymbol{y}_t - \boldsymbol{\Phi}_t\boldsymbol{s}\|_2^2 + \kappa\widetilde{\mathcal{V}}_\mu(\boldsymbol{s}, \boldsymbol{F}_t\widehat{\boldsymbol{s}}_{t-1}), \tag{4.7}$$

where parameter $\kappa > 0$ balances between data fidelity (first term) and dynamics (second term). Here, the dynamical prior is the previous estimate $\widehat{\boldsymbol{s}}_{t-1} \in \mathbb{R}^N$ of this estimation problem, propagated in time via a linear dynamical process $\boldsymbol{F}_t$.

Since it is crucial to understand the behavior and efficacy of unbalanced OT vis-a-vis other OT models, we begin by studying different types of OT-regularization in regimes where the total mass changes across video frames. OT-regularization depends on principles of mass-conservation, thus, inference is negatively affected when such assumptions are violated. We simplify our study by considering the problem of single-frame denoising with an OT-regularizer

$$\min_{\boldsymbol{s}} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{s}\|_2^2 + \kappa \mathcal{T}(\boldsymbol{s}, \boldsymbol{s}_0),$$

where $\boldsymbol{y}$ is a noisy observation, $\boldsymbol{s}_0$ is a dynamical-prior (with an identity dynamical function, i.e., $\boldsymbol{F}_t = \boldsymbol{I}$), and the OT-regularizer $\mathcal{T}$ here takes one of three different OT strategies:

1. balanced-OT ($\mathcal{T} = \mathcal{W}$) according to (OT) which assumes that the total mass is equal to that of the prior-frame,

2. partial-OT ($\mathcal{T} = \mathcal{U}_\mu$) according to (4.1) which assumes that the total mass is the lesser of the total mass between the inferred-frame and the prior-frame, and

3. unbalanced-OT ($\mathcal{T} = \mathcal{V}_\mu$) according to (4.5) which assumes total amount of transported mass is regulated by an ℓ_p^p-norm penalty on growth/decay of mass, with $p = 1$ in this experiment.

We set up a simple illustrative example denoising simulation in which sparse 10×10-pixel images have total mass that changes from one frame to another, under two regimes: mass growth and mass decay. The goal is to recover the second frame $\boldsymbol{s}$ from a noisy observation $\boldsymbol{y} = \boldsymbol{s} + \boldsymbol{\eta}$, where $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$, with $\sigma = 0.1$. To simplify our empirical analysis, we assign $\boldsymbol{s}_0$ to be the (uncorrupted) first frame. Under the growth regime, active pixels change from a magnitude of 0.5 to 1.0, while under the decay regime, active pixels change from a magnitude of 1.5 to 1.0. In each of the two frames, spatial support movement of active pixels are randomly assigned according to a radial Gaussian probability whose mean is one pixel away from its original location. All algorithms in this experiment were

implemented with the CVX optimization software [128, 129] and solved using interior-point methods to ensure that they were reliably solved. In all methods, optimal parameters were found via exhaustive search.

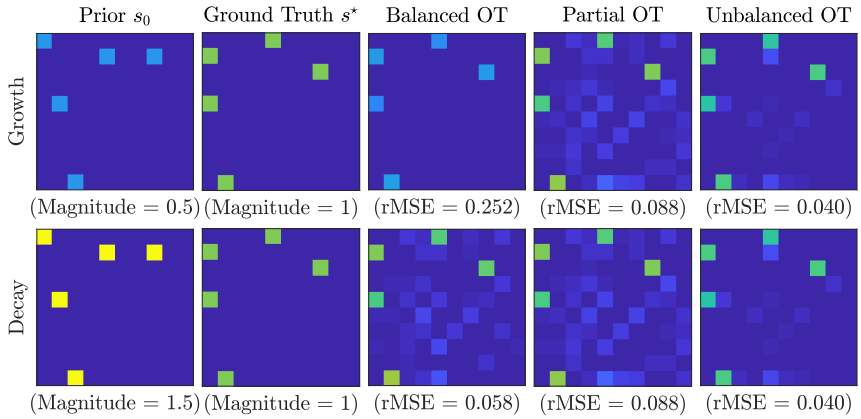


Figure 4.5: Qualitative reconstruction performance of OT-regularization schemes under mass changing regimes. We observe the optimal reconstruction solutions from denoising with various OT-priors under regimes of mass-growth (top row) and mass-decay (bottom row). Across both regimes, unbalanced-OT offers solutions with the lowest reconstruction error and consistently good support estimation.

First, we inspect the qualitative behavior of the various regularizers in Figure 4.5. Under a growth regime, Balanced-OT transports insufficient mass from $\boldsymbol{s}_0$, resulting in $\boldsymbol{s}$'s signal amplitude being grossly under-estimated. Conversely, under the decay regime, too much mass is transported so excess mass has to "overflow" into neighboring regions. In the growth regime, support estimation is interestingly perfect due to the limited transport budget, while support estimation is poor in the decay regime due to excess transport budget. For partial-OT, identical solutions are produced under both regimes and its reconstruction error is smaller than that of balanced-OT, but its support estimation is poor in both regimes. For unbalanced-OT, we observe similar solutions (though not identical) under both regimes. Compared to previous methods, it also has good support estimation and has the best overall reconstruction error. Similar to partial-OT, unbalanced-OT has a mechanism that adjusts the total transported mass (via parameter μ) so it does not suffer from mass overflow or underflow issues of balanced-OT. Furthermore, unbalanced-OT enjoys favorable support

estimation because it allows mass growth/decay at the individual pixel level.

In Figure 4.6, we quantitatively compare the reconstruction error of the different OT-regularizers. Reconstruction error is measured using *relative MSE* (rMSE) computed as $\|\widehat{\boldsymbol{s}} - \boldsymbol{s}^\star\|_2^2 \,/\, \|\boldsymbol{s}^\star\|_2^2$ for a given estimation $\widehat{\boldsymbol{s}}$ and ground truth $\boldsymbol{s}^\star$; rMSE lies in the range $[0, \infty]$ with 0 representing perfect reconstruction. In the first set of plots (top row) we vary noise levels (under a fixed mass-change rate of 0.5), and in the second set of plots (bottom row) we vary mass-change rates (under a fixed noise level of $\sigma = 0.1$). In very high-noise regimes, balanced-OT provides the best performance due to its strong mass-conservation assumptions. Yet, in typical noise regimes (e.g., $\sigma \leq 0.1$), partial-OT and unbalanced-OT is significantly advantageous compared to balanced-OT, with unbalanced-OT being consistently achieving the best performance. At various mass-change rates (under a fixed noise level of $\sigma = 0.1$), it is unsurprising that the higher the rate of mass change, the worse the balanced-OT's reconstruction error is. It however is interesting that partial- and unbalanced-OT's performance is quite invariant to rate of mass change, with unbalanced-OT being the superior of the two.

4.4 Scalable proximal primal-dual algorithm for an unbalanced OT regularizer

The proximal operator $\operatorname{prox}^{\mu}_{\rho\widetilde{\mathcal{V}}} : (\mathbb{R}^N \times \mathbb{R}^N) \mapsto (\mathbb{R}^N \times \mathbb{R}^N)$ of $\widetilde{\mathcal{V}}_\mu$ is defined as

$$\begin{aligned}
\operatorname{prox}^{\mu}_{\rho\widetilde{\mathcal{V}}}(\boldsymbol{p}_0, \boldsymbol{p}_1) &= \underset{\boldsymbol{x}_0, \boldsymbol{x}_1 \geq 0}{\arg\min}\ \widetilde{\mathcal{V}}_\mu(\boldsymbol{x}_0, \boldsymbol{x}_1) + \frac{1}{2\rho} \left\| \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \end{bmatrix} - \begin{bmatrix} \boldsymbol{p}_0 \\ \boldsymbol{p}_1 \end{bmatrix} \right\|_2^2 \\
&= \underset{\boldsymbol{x}_0, \boldsymbol{x}_1 \geq 0, \boldsymbol{M}, \boldsymbol{r}}{\arg\min}\ \|\boldsymbol{M}\|_{2,1} + \mu \|\boldsymbol{r}\|_p^p + \frac{1}{2\rho} \left\| \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \end{bmatrix} - \begin{bmatrix} \boldsymbol{p}_0 \\ \boldsymbol{p}_1 \end{bmatrix} \right\|_2^2 \\
&\qquad \text{s.t. } \operatorname{div}(\boldsymbol{M}) - \boldsymbol{x}_0 + \boldsymbol{x}_1 = \boldsymbol{r}.
\end{aligned} \tag{4.8}$$

This objective is strongly convex and not everywhere infinite, so according to proximal operator theory it has a unique minimizer for every $(\boldsymbol{p}_0, \boldsymbol{p}_1)$. Recent work [53] demon-

strated that Chambolle-Pock's first order primal-dual method [24] efficiently evaluates the *balanced* OT problem (2.5). We extend this result based on the same method and develop an efficient iterative algorithm to compute the *proximal operator* of the *unbalanced* OT problem (4.8). In addition, we provide a convergence guarantee.

The Lagrangian of (4.8) is:

$$\begin{aligned}\mathcal{L}(\boldsymbol{M}, \boldsymbol{r}, \boldsymbol{x}, \boldsymbol{a}) = \ & \|\boldsymbol{M}\|_{2,1} + \mu\|\boldsymbol{r}\|_p^p + \frac{1}{2\rho}\|\boldsymbol{x} - \boldsymbol{p}\|_2^2 \\ & + \iota_+(\boldsymbol{x}) + \langle \boldsymbol{a}, \mathrm{div}(\boldsymbol{M}) + \boldsymbol{A}\boldsymbol{x} - \boldsymbol{r}\rangle\,,\end{aligned} \tag{4.9}$$

where $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x}_0 \\ \boldsymbol{x}_1\end{bmatrix}$, $\boldsymbol{p} = \begin{bmatrix}\boldsymbol{p}_0 \\ \boldsymbol{p}_1\end{bmatrix}$, $\boldsymbol{A} = [-\boldsymbol{I}, \boldsymbol{I}]$, ι_+ is an indicator function of the non-negative orthant, and $\boldsymbol{a} \in \mathbb{R}^N$ is a Lagrange multiplier. The saddle point to

$$\min_{\boldsymbol{M},\boldsymbol{x},\boldsymbol{r}} \max_{\boldsymbol{a}} \ \mathcal{L}(\boldsymbol{M}, \boldsymbol{r}, \boldsymbol{x}, \boldsymbol{a}) \tag{4.10}$$

solves (4.8), for which the primal-dual method of Chambolle and Pock generates the following convergent sequence:

$$\boldsymbol{M}^{(t+1)} \leftarrow \arg\min_{\boldsymbol{M}} \ \|\boldsymbol{M}\|_{2,1} + \langle \boldsymbol{a}^{(t)}, \mathrm{div}(\boldsymbol{M})\rangle + \frac{1}{2\tau_1}\|\boldsymbol{M} - \boldsymbol{M}^{(t)}\|_F^2 \tag{4.11}$$

$$\boldsymbol{x}^{(t+1)} \leftarrow \arg\min_{\boldsymbol{x}\geq 0} \ \frac{1}{2\rho}\|\boldsymbol{x} - \boldsymbol{p}\|_2^2 + \langle \boldsymbol{a}^{(t)}, \boldsymbol{A}\boldsymbol{x}\rangle + \frac{1}{2\tau_1}\|\boldsymbol{x} - \boldsymbol{x}^{(t)}\|_2^2 \tag{4.12}$$

$$\boldsymbol{r}^{(t+1)} \leftarrow \arg\min_{\boldsymbol{r}} \ \mu\|\boldsymbol{r}\|_p^p + \langle \boldsymbol{a}^{(t)}, -\boldsymbol{r}\rangle + \frac{1}{2\tau_1}\|\boldsymbol{r} - \boldsymbol{r}^{(t)}\|_2^2 \tag{4.13}$$

$$\boldsymbol{a}^{(t+1)} \leftarrow \arg\max_{\boldsymbol{a}} \ \langle \boldsymbol{a}, \boldsymbol{b}^{(t+1)}\rangle - \frac{1}{2\tau_2}\|\boldsymbol{a} - \boldsymbol{a}^{(t)}\|_2^2\,, \tag{4.14}$$

where $\boldsymbol{b}^{(t+1)} = 2\mathcal{K}(\boldsymbol{M}^{(t+1)}, \boldsymbol{x}^{(t+1)}, \boldsymbol{r}^{(t+1)}) - \mathcal{K}(\boldsymbol{M}^{(t)}, \boldsymbol{x}^{(t)}, \boldsymbol{r}^{(t)})$ and $\mathcal{K}(\boldsymbol{M}, \boldsymbol{x}, \boldsymbol{r}) = \mathrm{div}(\boldsymbol{M}) +$

$\boldsymbol{A}\boldsymbol{x} - \boldsymbol{r}$.

Updates (4.11)-(4.14) all have standard closed-form proximal algorithms [27]. For (4.11), we exploit the fact that the program is row-wise separable. Denoting each row of $\boldsymbol{M}^{(t+1)}$ as $\boldsymbol{m}_i^{(t+1)} \in \mathbb{R}^2, \forall i = \{1, \ldots, N\}$, we apply the $\ell_{2,1}$-norm proximal algorithm (vector-soft shrinkage operator) on each of its rows:

$$\boldsymbol{m}_i^{(t+1)} = \text{shrink}_{\tau_1}^{\ell_2}(\boldsymbol{m}_i^{(t)} - \tau_1 \text{div}^*(\boldsymbol{a}^{(t)})_i), \tag{4.15}$$

where $\text{div}^*(\cdot) : \mathbb{R}^N \mapsto \mathbb{R}^{N\times 2}$ refers to the adjoint of the $\text{div}(\cdot)$ operator and where $\text{shrink}_{\sigma}^{\ell_2}(\boldsymbol{q}) = \max\{\|\boldsymbol{q}\|_2 - \sigma, 0\} \odot \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|_2}$. Next, (4.12) applies a projection unto the non-negative orthant:

$$\boldsymbol{x}^{(t+1)} = \Pi_+\Big(\frac{\rho\tau_1}{1+\rho\tau_1}\boldsymbol{p} + \frac{1}{1+\rho\tau_1}\big(\boldsymbol{x}^{(t)} - \tau_1 \boldsymbol{A}^\top \boldsymbol{a}^{(t)}\big)\Big), \tag{4.16}$$

where $\Pi_+(\boldsymbol{q}) = \max\{\boldsymbol{q}, 0\}$. For (4.13), we let $p = 1$ here to apply a linear penalty on mass growth/decay. Applying an ℓ_1 shrinkage operator yields:

$$\boldsymbol{r}^{(t+1)} = \text{shrink}_{\mu\tau_1}^{\ell_1}(\boldsymbol{r}^{(t)} + \tau_1 \boldsymbol{a}^{(t)}), \tag{4.17}$$

where $\text{shrink}_{\sigma}^{\ell_1}(\boldsymbol{q}) = \text{sign}(\boldsymbol{q}) \odot \max\{|\boldsymbol{q}| - \sigma, 0\}$. We remark that the update step for $\boldsymbol{r}$ with $p = 2$ may be alternatively derived to be an averaging update. Lastly, (4.14) is a simple projection:

$$\boldsymbol{a}^{(t+1)} = \boldsymbol{a}^{(t)} + \tau_2 \boldsymbol{b}^{(t+1)}. \tag{4.18}$$

The full algorithm is presented in Algorithm 3. Although this proximal algorithm is itself an iterative procedure, its update steps are very efficient when implemented on distributed computing since they have element-wise closed form solutions. Moreover, when used as a part of a larger solver, warm-starts and early-termination can significantly reduce the number of proximal iterations, as we will demonstrate in Section 4.4.2. We note that this method was found to be very efficient at solving an equality-constrained subproblem, compared to

an alternative strategy involving a series of affine projections [27, §6.2.2].

Algorithm 3 Unbalanced Beckmann OT Proximal Algorithm.

Require: $\boldsymbol{M}^{(0)}$, $\boldsymbol{x}^{(0)}$, $\boldsymbol{r}^{(0)}$, $\boldsymbol{a}^{(0)}$, μ, ρ, τ_1, τ_2
Ensure: $\boldsymbol{x}^{(t)}$

$k = 1$
while not converged **do**
  $\boldsymbol{m}_i^{(t+1)} = \mathrm{shrink}_{\tau_1}^{\ell_2}(\boldsymbol{m}_i^{(t)} - \tau_1 \mathrm{div}^*(\boldsymbol{a}^{(t)})_i), \ \ \forall i$
  $\boldsymbol{x}^{(t+1)} = \Pi_+\big(\frac{\rho\tau_1}{1+\rho\tau_1}\boldsymbol{p} + \frac{1}{1+\rho\tau_1}(\boldsymbol{x}^{(t)} - \tau_1 \boldsymbol{A}^\top \boldsymbol{a}^{(t)})\big)$
  $\boldsymbol{r}^{(t+1)} = \mathrm{shrink}_{\mu\tau_1}^{\ell_1}(\boldsymbol{r}^{(t)} + \tau_1 \boldsymbol{a}^{(t)})$
  $\boldsymbol{a}^{(t+1)} = \boldsymbol{a}^{(t)} + \tau_2 \boldsymbol{b}^{(t+1)}$
end while

We conclude this section with an analytic guarantee that specifies step size conditions for the convergence of Algorithm 3 to the saddle point of (4.10).

Theorem 4.4.1 (Convergence guarantee)**.** *Suppose* $\tau_1\tau_2 < \frac{1}{\lambda_{\max}(\nabla^2)+3}$ *where* $\lambda_{\max}(\nabla^2)$ *is the largest eigenvalue of the discrete Laplacian operator,* ∇^2*. Then the steps in Algorithm 1 produce a series which converges to the saddle point of the Lagrangian, i.e.,*

$$(\boldsymbol{M}^k, \boldsymbol{r}^k, \boldsymbol{x}^k, \boldsymbol{a}^k) \to (\boldsymbol{M}^\star, \boldsymbol{r}^\star, \boldsymbol{x}^\star, \boldsymbol{a}^\star),$$

where $(\boldsymbol{M}^\star, \boldsymbol{r}^\star, \boldsymbol{x}^\star, \boldsymbol{a}^\star)$ *is a solution of* (4.10)*.*

Please refer to Appendix A for the proof.

4.4.1 OT-regularized robust PCA: foreground/background separation in video

For our second application, we consider an ill-posed joint separation and inverse problem, where the goal is to separate a superposition of signal $\boldsymbol{S}$ and clutter (or background/interference) $\boldsymbol{L}$ embedded in noisy measurements $\boldsymbol{Y}$. A similar dynamical propagation model (as previously described) is assumed between consecutive frames of $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_T] \in \mathbb{R}^{N\times T}$, while clutter is denoted with $\boldsymbol{L} = [\boldsymbol{l}_1, \ldots, \boldsymbol{l}_T] \in \mathbb{R}^{N\times T}$ where $\mathrm{rank}(\boldsymbol{L}) = R \leq \min(N, T)$.

Measurements $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T] \in \mathbb{R}^{M \times T}$ are captured by the model

$$\boldsymbol{y}_t = \boldsymbol{\Phi}_t(\boldsymbol{s}_t + \boldsymbol{l}_t) + \boldsymbol{\eta}_t, \qquad t = 1, \ldots, T,$$

where $\boldsymbol{\Phi}_t \in \mathbb{R}^{M \times N}$ is the measurement matrix, and $\boldsymbol{\eta}_t \in \mathbb{R}^M$ denotes iid Gaussian measurement noise. Unlike the online nature of (4.7), this application requires a batch of time-windowed data to capture sufficient information so that clutter can be differentiated from the sparse signals. To solve this problem, we employ a framework called *robust principal components analysis (RPCA)* to separate $\boldsymbol{S}$ from $\boldsymbol{L}$, and augment it using our unbalanced OT model to exploit dynamical continuity between the adjacent frames of $\boldsymbol{S}$.

Before proceeding, we describe some RPCA background. RPCA separates data $\boldsymbol{X}$ comprising of a superposition of sparse outliers $\boldsymbol{S}$ and a low-rank component $\boldsymbol{L}$ via

$$\min_{\boldsymbol{S},\boldsymbol{L}} \ \|\boldsymbol{S}\|_1 + \gamma\|\boldsymbol{L}\|_* \ \text{ s.t. } \ \boldsymbol{X} = \boldsymbol{S} + \boldsymbol{L}, \tag{4.19}$$

where $\|\boldsymbol{S}\|_1 = \sum_{ij} |S_{ij}|$ and $\|\boldsymbol{L}\|_*$ refers to the nuclear norm, i.e., sum of magnitudes of $\boldsymbol{L}$'s singular values, with some parameter $\gamma > 0$. It was shown in [139, 151] that under incoherence and randomness conditions on $\boldsymbol{S}$ and $\boldsymbol{L}$, solving (4.19) with parameter $\gamma = \sqrt{\max(N, T)}$ recovers $\boldsymbol{S}$ and $\boldsymbol{L}$ with high probability, provided $\boldsymbol{S}$ is sufficiently sparse and $\boldsymbol{L}$ is sufficiently low-rank. A relevant application to this work is video surveillance [139], where video frames are packed into the columns of $\boldsymbol{X}$. In this scenario, $\boldsymbol{L}$ corresponds to the stationary background while $\boldsymbol{S}$ captures moving objects in the foreground. A rich literature has developed around the idea of sparse and low-rank decompositions [152]. One branch of work focuses on enforcing additional structure on the sparse component to encourage solutions that vary continuously over time, for example by employing optical flow based methods [153] or Markov Random Fields [154]. However, these methods assume full access to the original video frames and would require non-trivial adaptations to allow for recovery from compressive measurements.

In this work, we extend the stable formulation of RPCA [155] with compressive measurements [156]

$$\min_{\boldsymbol{S},\boldsymbol{L}\geq 0} \ \frac{1}{2}\sum_{t=1}^{T}\left(\|\boldsymbol{y}_t-\boldsymbol{\Phi}_t(\boldsymbol{s}_t+\boldsymbol{l}_t)\|_2^2\right)+\lambda\|\boldsymbol{S}\|_1+\gamma\|\boldsymbol{L}\|_*, \tag{4.20}$$

and show how our unbalanced OT model can easily be incorporated to use optimal transport as a continuity regularizer on the sparse component. Specifically, we propose robust PCA with UOT-regularized dynamic-filtering (RPCA+UOT-DF):

$$\min_{\boldsymbol{S},\boldsymbol{L}\geq 0} \ \frac{1}{2}\sum_{t=1}^{T}\left(\|\boldsymbol{y}_t-\boldsymbol{\Phi}_t(\boldsymbol{s}_t+\boldsymbol{l}_t)\|_2^2\right)+\lambda\|\boldsymbol{S}\|_1+\gamma\|\boldsymbol{L}\|_*+\kappa\sum_{t=1}^{T-1}\widetilde{\mathcal{V}}_\mu(\boldsymbol{s}_t,\boldsymbol{s}_{t+1}). \tag{4.21}$$

The first term is a data fidelity term, the second term a sparsity prior, the third term a low-rank prior, and the last term the unbalanced OT regularizer that promotes temporal coherence across the sparse frames of the signal. Applying the theory in [151] allows us to reduce one parameter due to the relation $\gamma/\lambda=\sqrt{\max(N,T)}$.

4.4.2 Efficiency of proximal Beckmann in ADMM implementation of RPCA+UOT-DF

Proximal algorithms have maximal utility when they can be computed efficiently (i.e., closed form solutions at best, and iterative at worst). Although the proposed Beckmann proximal algorithm is iterative, we submit that it can still be extremely efficient by applying standard strategies such as:

- *warm starts* – instead of restarting the Beckmann proximal algorithm at each ADMM iteration, we warm-start it using its state from the previous ADMM iteration, and
- *inexact updates* (early termination) – rather than solving the proximal algorithm to high precision, we only partially solve it by terminating it early after a fixed (predetermined) number of iterations.

To empirically demonstrate the benefits of the above strategies, we apply them to our pro-

posed Beckmann proximal algorithm within the ADMM framework of the RPCA+UOT-DF solver described in Section 4.4.1. In Figure 4.7 we inspect the computational complexity associated with ADMM as a function of solution exactness (i.e., accuracy) of the Beckmann proximal algorithm. Exactness here is implied by the number of fixed iterations we let it run (i.e., more iterations imply a more exact solution). To measure the computational complexity of ADMM, we count the number of ADMM iterations required for it satisfy the following stopping criterion: the primal and dual residuals (defined [22, §3.3]) must both reach a value less than 10^{-4}. *Additional % of ADMM iterations* (y-axis) is defined as $|b(k) - b(30)|/b(30)$, where $b(k)$ refers to the recorded number of ADMM iterations at convergence, while k refers to the number of fixed Beckmann proximal iterations (x-axis). We use $b(30)$ as our baseline since convergence of the proximal algorithm is observed at this value of k. Unsurprisingly, our results demonstrate that that ADMM's computational complexity decays as a function of proximal solution exactness. Remarkably however, even at the upperbound (a single proximal iteration), the overall ADMM complexity is inflated by merely 1%, suggesting that our proximal algorithm can be very efficient when used with these strategies.

4.4.3 RPCA+UOT-DF quantitative performance on synthetic data

In this section, we characterize RPCA+UOT-DF ($p = 1$) on synthetic data, which is generated with the following procedure. The matrix $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_T]$ is a sparse matrix whose columns $\boldsymbol{s}_t \in \mathbb{R}_+^N$ are vectorized $n \times n$ images ($N = n^2$). We begin by generating K active pixels (targets) in $\boldsymbol{s}_1$ then, for each consecutive frame, randomly move them in similar fashion as described in Section 4.3.1. The top row of Figure 4.8 illustrates how the ground truth is generated on a simple dataset with 10×10 pixels $\times$ 6 frames. $\boldsymbol{L} = [\boldsymbol{l}_1, \ldots, \boldsymbol{l}_T]$ is generated by multiplying two low-rank matrices $\boldsymbol{U}\boldsymbol{V}^\top/4R$ where $\boldsymbol{U} \in \mathbb{R}_+^{N \times R}$ and $\boldsymbol{V} \in \mathbb{R}_+^{T \times R}$ are matrices of rank $R \leq \min(N, T)$, and whose entries are distributed as $U_{ij}, V_{ij} \sim \text{Uniform}(0, 1)$. $\boldsymbol{\Phi}_t \in \mathbb{R}^{M \times T}$ have entries that are randomly generated from an

iid normal distribution with a variance of $1/M$. The observation matrix $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T]$ consists of columns produced by $\boldsymbol{y}_t = \boldsymbol{\Phi}_t(\boldsymbol{s}_t + \boldsymbol{l}_t) + \boldsymbol{\eta}_t$, with noise $\boldsymbol{\eta}_t \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$.

In our experiments, we compared against three other algorithms: (i) robust PCA (RPCA) [139] to serve as a benchmark, (ii) RPCA with a *balanced-OT* regularizer (RPCA+BOT-DF), and (iii) RPCA with an ℓ_1 dynamical filter (RPCA+L1-DF) as a cheap dynamical filtering alternative to OT methods. For this solver, we replace the unbalanced-OT prior in (4.21) with $\sum_{t=1}^{T-1} \|b(\boldsymbol{s}_t) - b(\boldsymbol{s}_{t+1})\|_1$ where $b(\cdot)$ refers to a Gaussian blurring convolution operator with a 3×3 kernel since we do not expect targets to travel too far in this simulation. For RPCA+BOT-DF, we replace the unbalanced-OT prior in (4.21) with a *balanced-OT* prior (4.6). All methods are implemented in ADMM [22] with the following stopping criteria: primal and dual residuals must both be $\leq 10^{-4}$ or a maximum of 5000 iterations are reached.

In terms of performance metrics we used: (i) *relative MSE* (rMSE) to measure the normalized ℓ_2 reconstruction error, and (ii) *F1 score* to measure accuracy of estimated support. The F1 score (also called the *Sørensen-Dice coefficient*) is computed as the harmonic mean between precision and recall of reconstruction and ground truth masks (using a threshold set at an intensity of 0.05) and computed as $2(\|\boldsymbol{m}^\star \cap \widehat{\boldsymbol{m}}\|_0)/(\|\boldsymbol{m}^\star\|_0 + \|\widehat{\boldsymbol{m}}\|_0)$, where $m_i = 1$ if $x_i \geq$ threshold and 0 otherwise, and $\|\cdot\|_0$ refers to the cardinality of the argument. F1 score lies between 0 and 1, with 1 representing perfect support estimation. Experimental plots show aggregated results from 20 randomly generated trials, with the markers displaying the median and error bars denoting the 25^{th} to 75^{th} quantiles.

Figure 4.8 qualitatively compares the application of our algorithm against the benchmark RPCA. RPCA generally experiences more noise throughout the reconstruction, and occasionally gross corruption (e.g., frames 1 and 5). RPCA+L1-DF's sparsity prior on frame differences provides some advantage in support estimation (frame 3 is improved compared to RPCA), yet it is still prone to gross errors (frame 5). RPCA+BOT-DF similarly shows improvement over RPCA, however we observe mass-overflow effects that were

previously also observed in Section 4.3.1. In contrast, RPCA+UOT-DF demonstrates remarkable robustness and demonstrates significantly improved support estimation over the other algorithms.

Next, we run a series of experiments that characterize the performance of the proposed algorithm in comparison to other algorithms. To this end, the following simulation parameters are varied: (i) the Gaussian noise in the measurements, (ii) the compression ratio M/N of the measurement matrix $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$, (iii) the number of sparse targets K (iv) the rank of the low-rank interference component $\boldsymbol{L}$, and (v) the number of frames in a batch T. For each each simulation setting, all solvers are tuned by searching for the optimal parameters (e.g., λ, κ, μ) via a (logarithmic space) pattern search algorithm [157], with the low-rank weight assigned as $\gamma = \lambda\sqrt{\max(N, T)}$. To highlight the advantage of our algorithm, we selected a range of simulation parameter that demonstrates its superior performance. Unless explicitly varied in each experiment, our baseline parameters are: noise-level $\sigma = 0.001$, compression ratio $M/N = 0.6$, sparsity of $K = 5$ pixels, clutter rank of $R = 1$, and rate of mass change of 1 over the intensity range of 0.5 to 1.5. In figure 4.9, we see that the proposed algorithm has superior performance over the other algorithms over all three metrics, especially in its ability to accurately recover the support. In the first column of plots, we observe that unbalanced-OT is most effective when noise is low ($\sigma < 0.1$), since reconstruction suffers when noise rather than signal is the subject of transportation. In the second column of plots, we observe a region of significant support accuracy advantage ($M/N < 0.5$) despite marginal rMSE advantage over other algorithms. The third column of plots showcases unbalanced-OT's advantage in sparse regimes: it exploits sparsity to register targets and provide regularization structure. In the fifth column of plots, we observe that unbalanced-OT has superior sample complexity, since dynamical correlations are more accurately described by our model. Finally, we see that although RPCA+BOT-DF and RPCA+UOT-DF shares similar formulations, and similar solve times (fifth row), the simple unbalanced modification obtains a significant performance gain.

4.4.4 RPCA+UOT-DF qualitative performance on real data

Finally, we evaluate performance on a video sequence of a person walking through an indoor scene. The snippet consists of 2 seconds of footage recorded at 30 frames per second and is downsampled to a resolution of 95×160 pixels ($T = 60$ and $N = 15,200$). In such practical applications, the foreground component may be darker or lighter than the background, so we must modify the RPCA+UOT-DF formulation to remove the $\boldsymbol{S} \geq 0$ constraint. Since the OT formulation takes nonnegative signals as its inputs, we decompose the sparse component into positive and negative components (in similar fashion as [158, 159, 160]) $\boldsymbol{S} = \boldsymbol{S}^+ - \boldsymbol{S}^-$ with $\boldsymbol{S}^+, \boldsymbol{S}^- \geq 0$ and add an OT regularization term for each component. The RPCA+UOT-DF objective function then becomes

$$\begin{aligned} \min_{\boldsymbol{S}^+,\boldsymbol{S}^-,\boldsymbol{L}\geq 0} \; & \frac{1}{2}\sum_{t=1}^{T}\left(\|\boldsymbol{y}_t - \boldsymbol{\Phi}_t(\boldsymbol{s}_t^+ - \boldsymbol{s}_t^- + \boldsymbol{l}_t)\|_2^2\right) + \lambda(\|\boldsymbol{S}^+\|_1 + \|\boldsymbol{S}^-\|_1) + \gamma\|\boldsymbol{L}\|_* \\ & + \kappa\sum_{t=1}^{T-1}\left(\widetilde{\mathcal{V}}_\mu(\boldsymbol{s}_t^+, \boldsymbol{s}_{t+1}^+) + \widetilde{\mathcal{V}}_\mu(\boldsymbol{s}_t^-, \boldsymbol{s}_{t+1}^-)\right). \end{aligned} \tag{4.22}$$

In these simulations, we observe severely compressed measurements ($M/N = 0.15$) and use RPCA (4.20), RPCA+L1-DF and RPCA+UOT-DF (4.22) to extract the moving person from the background scene. As before, pattern search was employed in the selection of algorithm parameters. However, to avoid the prohibitive computation time required to optimize directly with the full resolution data, parameters were chosen by first using pattern search on heavily downsampled data to obtain an approximation to the optimal parameter set, and then fine-tuned manually using the original data. Unlike previous simulations, we found that the relationship $\gamma = \lambda\sqrt{N}$ did not yield optimal results, so λ and γ were selected independently. Figure 4.10 shows several example frames which demonstrate how the UOT regularizer enables successful recovery even after compression. RPCA misses the foreground almost entirely, while RPCA+L1-DF yields only a crude estimate due to the inability of the ℓ_1 dynamics regularizer to effectively capture continuity in the sparse

component. Admittedly, a flaw with this method is that it may be unable to handle rapid changes in relative darkness/lightness between the foreground and background; for future work, richer OT models should therefore be developed to handle mass transfer between the positive and negative components.

4.5 Discussion

In this chapter, we propose novel regularizers based on Beckmann's formulation for inverse imaging problems. We highlight two important features of this OT regularizer. First, compared to other OT regularizers in the literature, it has key modelling features that are crucial for imaging applications such as the ability to describe mass changes. Second, we utilize the efficient Beckmann's reformulation to significantly alleviate the traditionally exorbitant costs associated with computing optimal transport. In addition, we propose a distributed proximal algorithm for this regularizer, so that it can be used in conjunction with proximal first order solvers for large-scale imaging applications. In terms of applications, we focused on the problem of target tracking in online and batched settings, since temporal and spatial correlations are well modelled using the proposed regularizer. We empirically characterize our method against other benchmarks on a synthetic dataset, and demonstrate superiority at reconstructing sparse dynamical signals. Lastly, we qualitatively demonstrate the efficacy of our method at recovering a moving subject from compressive measurements of real video data.

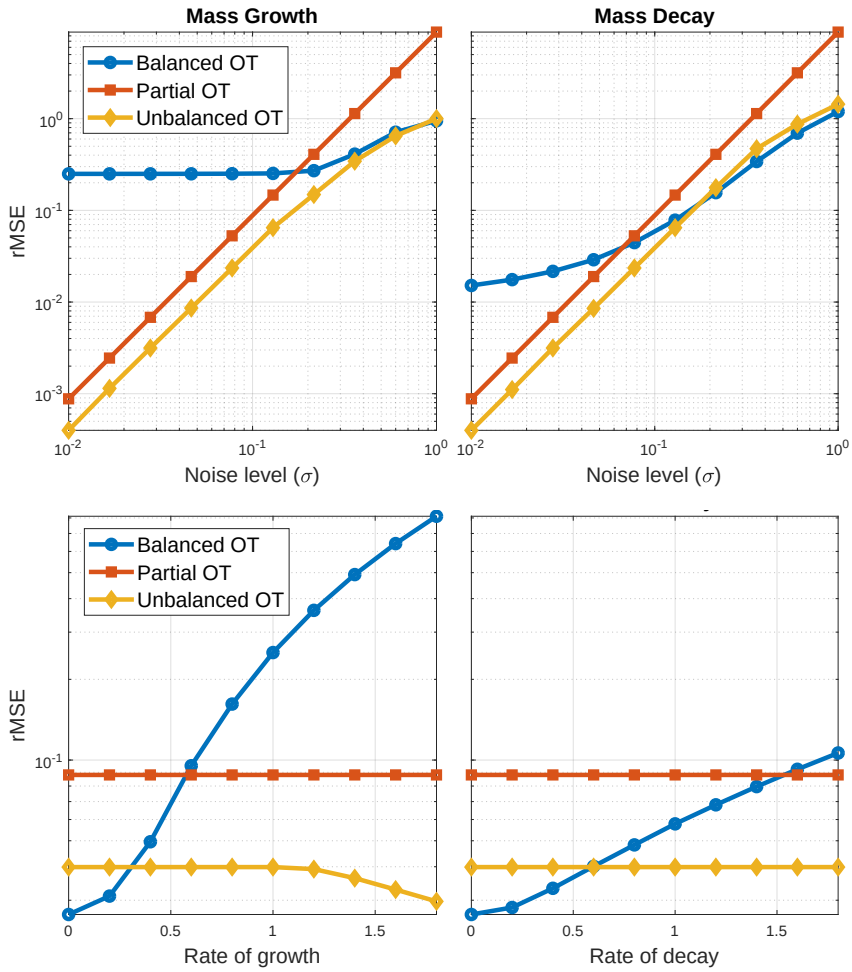


Figure 4.6: Quantitative reconstruction performance of OT-regularization schemes at varying noise levels (first row) and at varying growth/decay rates (second row). We subdivide the experiments into categories of mass growth (first column) and mass decay (second column). At various noise-levels, unbalanced-OT captures the best properties partial-OT and balanced-OT: at low-noise regimes it adopts partial-OT's low-error profile, while in high-noise regimes, it adopts the error profile of balanced-OT. At various mass change rates, we observe relatively stable performance in partial- and unbalanced-OT, while balanced-OT's performance suffers significantly as mass-change rate increases. Overall, unbalanced-OT has superior performance.

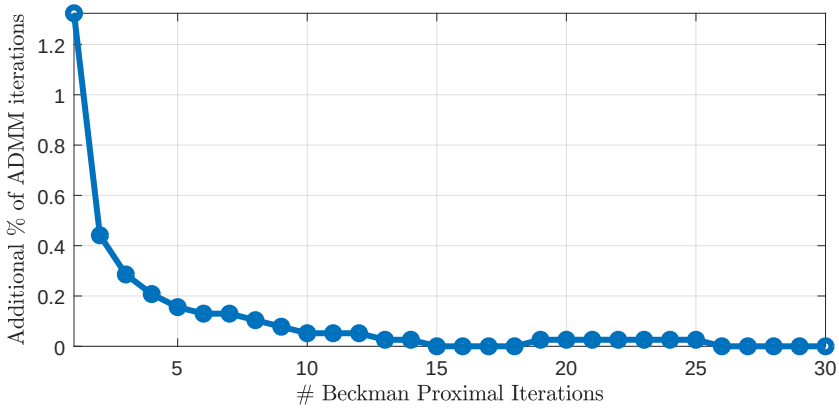


Figure 4.7: Additional ADMM computational cost (expressed as a percentage over the minimum possible number of ADMM iterations) as a function of proximal iterations. When warm-starts and early-termination strategies are applied, we see that even in worst case (just 1 proximal iteration), overall costs increase by merely 1%.

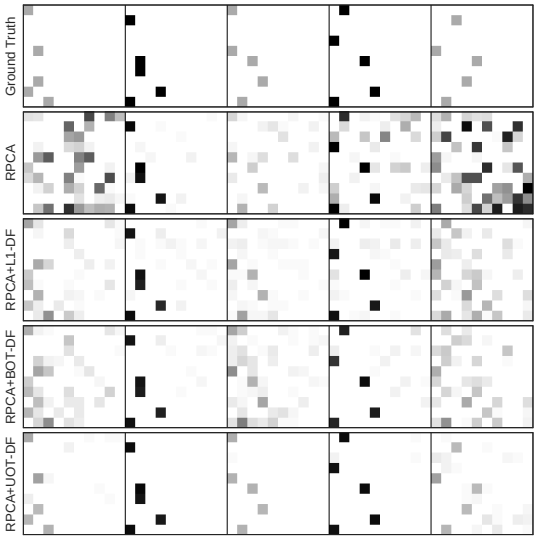


Figure 4.8: Qualitative example of reconstruction of sparse targets from compressive measurements. Row labels starting from top: ground truth, robust PCA's reconstruction, RPCA+L1-DF's reconstruction, RPCA+BOT-DF's reconstruction, (proposed) RPCA+UOT-DF's reconstruction. While robust PCA is quite successful at reconstructing the sparse targets, it is prone to severe error (frames 1, 3, 4, 5). Incorporating an optimal transport regularizer enforces temporal consistency across frames, significantly improving reconstruction performance, especially in accuracy of support.

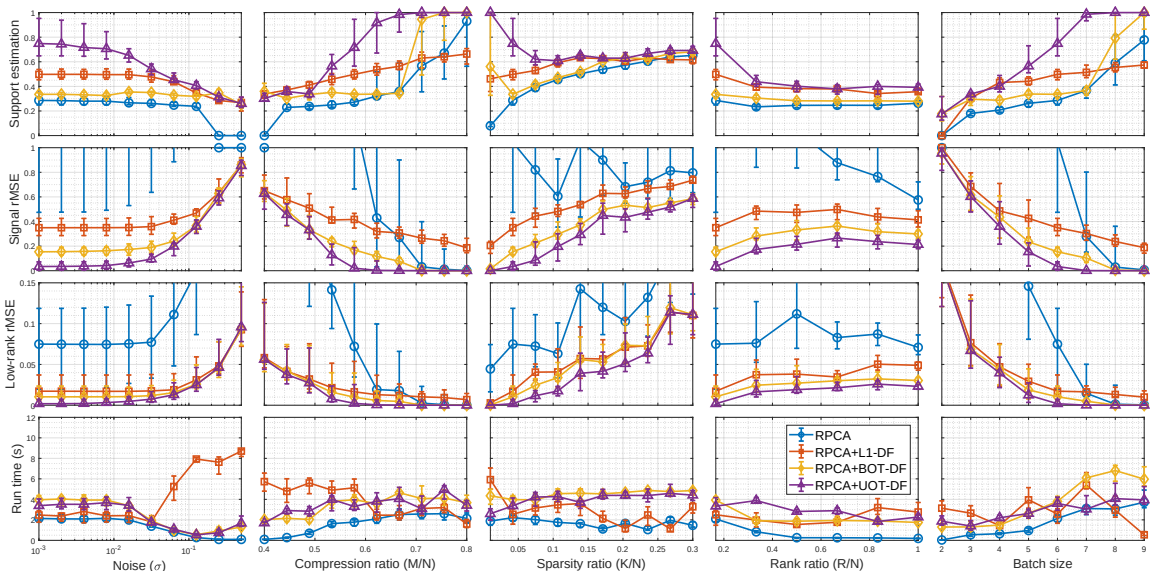


Figure 4.9: Performance of proposed algorithm (RPCA+UOT-DF) compared against other RPCA algorithms on synthetically generated data. To understand the algorithm's performance, we vary experimental simulation parameters such as: observation noise (first column), compression rate (second column), number of sparse targets (third column), rank of interference component (fourth column), number of frames in a batch (fifth column). The empirical performance limit of all algorithms is discovered by optimizing each algorithm to minimize reconstruction error from 20 independent random trials, as measured by its relative mean square error (rMSE). In all plots, we report the metric's trial median using a marker, while error bars denote the 25^{th} to 75^{th} quantiles. Our algorithm's main advantage is its significantly superior target support estimation (first row of plots), as measured using F1 score of target masks. The proposed algorithm also demonstrates superior performance under the metrics such as rMSE of the sparse component (second row), and rMSE of the low-rank component (third row). Finally, we note that algorithm's superior performance is obtained at comparable computational cost to the balanced-OT regularizer (fourth row).

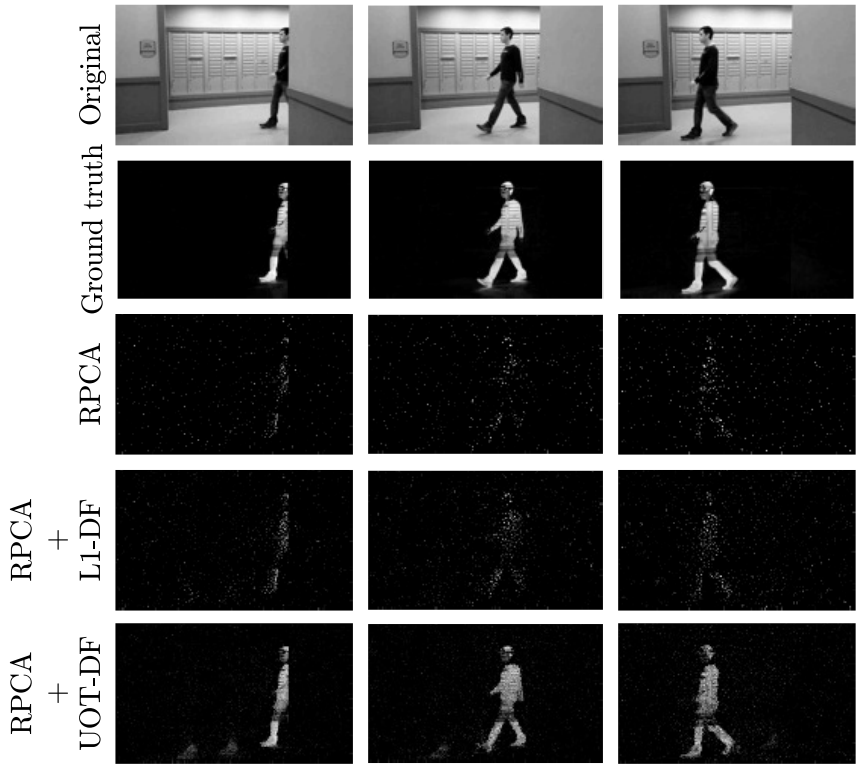


Figure 4.10: Separation of a moving subject from its environment using compressive measurements of a real video clip. Even under heavy compression ($M/N = 0.15$), the UOT regularizer enables nearly perfect recovery and separation of the person walking through the scene from the background. The foreground recovered by RPCA is noisy and the subject is all but lost. Although the dynamics regularizer in RPCA+L1-DF reduces noise somewhat, the ℓ_1 regularizer is not able to effectively leverage the continuity between frames and the subject remains barely distinguishable.

CHAPTER 5
EFFICIENT GENERALIZED OT REGULARIZATION FOR INVERSE PROBLEMS

In this chapter, we continue building efficient tools for optimal transport (OT) regularization in composite convex problems. Compared to the previous chapter that exploited the special ordering of Euclidean ground costs, here we extend our toolkit to more general settings of *arbitrary* ground costs, for which no such structure can be leveraged. We need fundamentally different tools and techniques since we are now optimizing directly over the transport coupling matrix. Computationally, this more challenging since it scales in $O(n^2)$ (n refers to the signal's dimension) compared to $O(n)$ of Beckmann methods of the previous chapter. Recently, Cuturi [63] introduced an efficient approximation to the OT problem using entropic regularization, which dramatically reduces the time complexity to *parallelizable* quadratic dependence on the signal's dimension. The key insight of entropic regularization is that the resulting algorithm (i.e., Sinkhorn iterations [64]) can be interpreted as iterative Bregman projections of a Gibbs density on an intersection of convex sets with respect to the Kullbeck-Liebler (KL) divergence [65]. This seminal work revived OT and ushered in a slew of theoretical and applied work in machine learning and the data sciences [43].

Despite promising progress, entropic regularization methods suffer several shortcomings. First, because mass-preservation is a key assumption in OT, most entropic regularization methods [63, 65, 141, 161, 56] cater specifically to probabilities (e.g., histograms in the simplex, or fixed-mass fluid systems). Recent works have attempted to generalize OT to signals in Euclidean space. For example, [142, 143] used *partial* OT [55] (i.e., transportation of a fraction of the total mass) to register moving targets across time-steps in time-varying inverse problems, and [45, 46] used OT to model deformations from signal priors in tomographic imaging applications. Although [46] generalized Sinkhorn iterations

for signals in Euclidean space, their method was subject to mass-preservation constraints, which is unrealistic in many applications. For instance in videos, the physical preservation of mass is likely to hold in 3 spatial dimensions across time, but is certainly violated when the scene is viewed from some 2-d projection plane; mass preservation constraints at the projection plane is therefore a poor model of reality. Recent works in *unbalanced* OT [57, 61, 56, 60] provide more generalizable OT models, yet the numerical methods for these techniques do not generalize well to Euclidean signals. Second, these methods explicitly trade-off accuracy for speed assuming that its application does not require high-precision solutions (e.g., in machine learning). This becomes problematic in applications (e.g., reconstruction) that are sensitive to the presence of entropic-approximation-artifacts such as blurring. Wang and Banerjee proposed a general solver framework called the *Bregman* alternating directions method of multipliers (BADMM) [36], which solved the OT problem free from entropic-approximation, via proximal Bregman iterations according to the KL divergence.

Inspired by recent advances in computational OT due to Bregman approaches [63, 43], we propose to fill the gaps surrounding the lack of tools that solve approximation-free variational OT problems for Euclidean signals. Specifically, we extend the flexible *Bregman* ADMM framework [36] to facilitate the inclusion of various types of OT-regularizers (e.g., balanced, or unbalanced) into the class of constrained convex problems in Euclidean space. The key idea of our method is to jointly account for *both* simplex and Euclidean geometries when taking gradient steps by using a composite Bregman proximal operator. Similar to popular entropic-regularization methods, our method enjoys parallelizable quadratic time-complexity in the signal's dimension, but is inherently free from approximation-artifacts caused by entropic regularization. Although we do not present new theoretical results, we show how the proposed framework integrates OT-regularization into convex inverse problems. In Section 5.1, we outline our composite Bregman ADMM framework, enumerate typical splitting use cases, and provide a host of building blocks in the form of

efficient closed form proximal algorithms for practical types of balanced and unbalanced OT-regularizers. In Section 5.2, we use an OT-regularized least-squares problem as an example to demonstrate (i) speed and memory advantages of a GPU implementation over a commercial interior point method, (ii) benefits in convergence rate over traditional ADMM [22], and (iii) accuracy advantages over a Sinkhorn method [46]. In Section 5.3, we illustrate the versatility and utility of our framework by including OT-regularization in two popular convex problems with novel applications: (i) the LASSO [3] for source localization from time-varying electroencephalogram (EEG) measurements, and (ii) robust PCA [139] for separation of signal from interference and noise in audio.

5.1 Efficient KL-Euclidean Bregman ADMM solver framework

In this section, we extend the Bregman ADMM framework [36] and customize it to the class of convex OT-regularized inverse problems. To begin, we formulate the generalized discrete OT-regularized convex problem as

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad J(\boldsymbol{x}) + \kappa\mathcal{W}(\mathcal{F}_1(\boldsymbol{x}), \mathcal{F}_2(\boldsymbol{x})), \tag{5.1}$$

where $J(\cdot)$ is a convex function (e.g., combination of data-fidelity term and regularization priors), $\kappa > 0$ a regularization weight, $\mathcal{W} : \mathbb{R}_+^q \times \mathbb{R}_+^q \mapsto \mathbb{R}_+$ a generic OT-regularizer, and $\mathcal{F}_1, \mathcal{F}_2 : \mathbb{R}^n \mapsto \mathbb{R}_+^q$ linear functionals that map $\boldsymbol{x}$ into the domain of non-negative OT marginals. To facilitate solver flexibility over various OT-regularizers, we decompose $\mathcal{W}$ into its transport cost and constraints:

$$\min_{\boldsymbol{x},\boldsymbol{P}} \quad J(\boldsymbol{x}) + \kappa \langle \boldsymbol{P}, \boldsymbol{C}\rangle + \iota_+(\boldsymbol{P}) + H(\mathcal{F}_1(\boldsymbol{x}), \mathcal{F}_2(\boldsymbol{x}), \boldsymbol{P}), \tag{5.2}$$

where $\boldsymbol{P} \in \mathbb{R}_+^{q\times q}$ is the transport coupling matrix, $\boldsymbol{C} \in \mathbb{R}_+^{q\times q}$ is the ground cost matrix, and H contains regularizer-specific constraints and functions.

5.1.1 OT-regularization variants

Here, we present two major OT variants that are well suited for inverse problems, along with modifications that allow them to fit into linearly constrained convex optimization problems. We describe these varieties with the modular function H defined in (5.2). Let $\boldsymbol{z} = \mathcal{F}_1(\boldsymbol{x})$ and $\widehat{\boldsymbol{z}} = \mathcal{F}_2(\boldsymbol{x})$ to ease notation.

Definition 5.1.1 (Balanced Transport)**.** *Balanced OT constraints in the original problem* (OT) *are described as*

$$H(\boldsymbol{z}, \widehat{\boldsymbol{z}}, \boldsymbol{P}) = \iota_{\boldsymbol{P}\mathbb{1}=\boldsymbol{z}}(\boldsymbol{z}, \boldsymbol{P}) + \iota_{\boldsymbol{P}^\top\mathbb{1}=\widehat{\boldsymbol{z}}}(\widehat{\boldsymbol{z}}, \boldsymbol{P}). \tag{BOT}$$

In unbalanced transport, we seek a balance between transport and mass growth/decay (through parameter $\mu > 0$). Compared to existing unbalanced schemes [61, 57], we may place non-negativity constraints on growth and decay to specify directionality, i.e., $\{\boldsymbol{P} \in \mathbb{R}_+^{n\times n} : \boldsymbol{P}\mathbb{1} \geq \boldsymbol{z}, \boldsymbol{P}^\top\mathbb{1} \geq \widehat{\boldsymbol{z}}\}$.

Definition 5.1.2 (Unbalanced Transport)**.** *Unbalanced OT constraints are described as*

$$\begin{aligned} H_\mu(\boldsymbol{z}, \widehat{\boldsymbol{z}}, \boldsymbol{P}) \;\; = \;\; & \mu(\|\boldsymbol{P}\mathbb{1} - \boldsymbol{z}\|_p^p + \|\boldsymbol{P}^\top\mathbb{1} - \widehat{\boldsymbol{z}}\|_p^p) + \iota_+(\boldsymbol{z}, \widehat{\boldsymbol{z}}) \\ & + \iota_{\boldsymbol{P}\mathbb{1}\geq\boldsymbol{z}}(\boldsymbol{z}, \boldsymbol{P}) + \iota_{\boldsymbol{P}^\top\mathbb{1}\geq\widehat{\boldsymbol{z}}}(\widehat{\boldsymbol{z}}, \boldsymbol{P}). \end{aligned} \tag{UOT}$$

For the Euclidean signal setting, $p = 1$ describes growth/decay profiles that are piecewise-smooth, while $p = 2$ describes growth/decay profiles that are graduating smoothly.

5.1.2 KL-Euclidean Bregman ADMM framework

We now describe our *variable splitting* method. The problem is reorganized such that $J + H \equiv f + \sum_i g_i$, where f contains primal variables $\boldsymbol{x}, \boldsymbol{P}$, and $\{g_i\}_i$ contains splitting

variables $\{\boldsymbol{z}_i, \boldsymbol{R}_i\}_i$. Equation (5.2) thus becomes

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{P},\{\boldsymbol{z}_i,\boldsymbol{R}_i\}_i} \; & f(\boldsymbol{x},\boldsymbol{P}) + \kappa \left\langle \boldsymbol{P}, \boldsymbol{C} \right\rangle + \iota_+(\boldsymbol{P}) + \sum_i g_i(\boldsymbol{z}_i, \boldsymbol{R}_i) \\ \text{s.t.} \quad & \mathcal{A}_i \boldsymbol{x} = \mathcal{B}_i \boldsymbol{z}_i, \quad \boldsymbol{P} = \boldsymbol{R}_i, \quad \forall i \end{aligned} \tag{5.3}$$

where $\{\mathcal{A}_i\}_i, \{\mathcal{B}_i\}_i$ are affine operators that implement $\mathcal{F}_1, \mathcal{F}_2$ in (5.1).

Our method is based on the fundamental observation that $\boldsymbol{x}$ and $\boldsymbol{P}$ belong to sets with inherently different geometries: $\boldsymbol{x}$ lies in Euclidean space while $\boldsymbol{P}$ lies in the set of doubly stochastic matrices. Since stochastic constraints (over $\boldsymbol{P}$) are a union of two sets of simplex constraints (i.e., over rows and over columns), we exploit this opportunity for splitting. Furthermore, [35] demonstrated that for the simplex set, the Bregman proximity that yields gradient steps with optimal convergence rates is the KL proximity. Since applying only one type of Bregman proximity (e.g., only KL or only Euclidean) would result in suboptimal descent on the other set, we propose a *composite* Bregman proximity term consisting of *both* quadratic ℓ_2 and KL divergences (for terms relating to $\boldsymbol{x}$ and $\boldsymbol{P}$, respectively):

$$B(\boldsymbol{x},\boldsymbol{P}|\boldsymbol{z},\boldsymbol{R}) \overset{\text{def.}}{=} \frac{1}{2}\left\|\boldsymbol{x}-\boldsymbol{z}\right\|_2^2 + \mathrm{KL}(\boldsymbol{P}|\boldsymbol{R}). \tag{5.4}$$

The resulting Bregman ADMM update steps are:

$$\begin{aligned} (\boldsymbol{x}^{(t+1)}, \boldsymbol{P}^{(t+1)}) \leftarrow \; & \arg\min_{\boldsymbol{x},\boldsymbol{P}} \; f(\boldsymbol{x},\boldsymbol{P}) + \kappa \left\langle \boldsymbol{P}, \boldsymbol{C} \right\rangle + \iota_+(\boldsymbol{P}) \\ & + \sum_i \left\langle \boldsymbol{a}_i^{(t)}, \mathcal{A}_i \boldsymbol{x} \right\rangle + \left\langle \boldsymbol{B}_i^{(t)}, \boldsymbol{P} \right\rangle + \rho B(\mathcal{A}_i \boldsymbol{x}, \boldsymbol{P} | \mathcal{B}_i \boldsymbol{z}_i^{(t)}, \boldsymbol{R}_i^{(t)}), \end{aligned} \tag{5.5}$$

$$\begin{aligned} (\boldsymbol{z}_i^{(t+1)}, \boldsymbol{R}_i^{(t+1)}) \leftarrow & \arg\min_{\boldsymbol{z}_i,\boldsymbol{R}_i} \; g_i(\boldsymbol{z}_i, \boldsymbol{R}_i) + \left\langle \boldsymbol{a}_i^{(t)}, -\mathcal{B}_i \boldsymbol{z}_i \right\rangle \\ & + \left\langle \boldsymbol{B}_i^{(t)}, -\boldsymbol{R}_i \right\rangle + \rho B(\mathcal{B}_i \boldsymbol{z}_i^{(t)}, \boldsymbol{R}_i^{(t)} | \mathcal{A}_i \boldsymbol{x}^{(t+1)}, \boldsymbol{P}^{(t+1)}), \quad \forall i, \end{aligned} \tag{5.6}$$

$$\begin{aligned}\boldsymbol{a}_i^{(t+1)} &\leftarrow \boldsymbol{a}_i^{(t)} + \tau(\mathcal{A}_i \boldsymbol{x}^{(t+1)} - \mathcal{B}_i \boldsymbol{z}_i^{(t+1)}), \ \ \forall i, \\ \boldsymbol{B}_i^{(t+1)} &\leftarrow \boldsymbol{B}_i^{(t)} + \tau(\boldsymbol{P}^{(t+1)} - \boldsymbol{R}_i^{(t+1)}), \ \ \forall i,\end{aligned} \tag{5.7}$$

where the superscript $(+)$ denotes the next ADMM iteration, $\{\boldsymbol{a}_i, \boldsymbol{B}_i\}_i$ are Lagrange dual variables, $\rho > 0$ is a proximal weight, and $\tau > 0$ is the dual ascent step size. When τ is sufficiently small, Bregman ADMM updates (5.5)-(5.7) for general Bregman divergences (including composite ones[1]) will converge to a stationary point of the sequence, in addition to having an overall linear convergence rate [36].

This method is computationally attractive for two reasons. First, this algorithm is insensitive to the dimension n of the problem due to geometric properties of the KL divergence w.r.t. simplex sets [35]; it converges more quickly than an quadratic ℓ_2 divergence by a factor of $O(n/\log n)$ [36]. Second, update steps (5.5)-(5.6) derived from our composite Bregman operators admit efficient closed form proximal algorithms. When elements in $\boldsymbol{x}$ and $\boldsymbol{P}$ (or $\boldsymbol{z}_i$ and $\boldsymbol{R}_i$) are separable, we may exploit standard first-order proximal approaches to solve for these variables, such as quadratic ℓ_2 proximity operators for $\boldsymbol{x}$ (e.g., [27, 162]), or exponentiated (entropic) proximal operators for $\boldsymbol{P}$ (e.g., [36, 65]). When elements are coupled (e.g., an indicator function with constraint $\boldsymbol{P}\mathbb{1} = \boldsymbol{x}$), we demonstrate in Section 5.1.4 that efficient closed form algorithms also exist for typical OT variants.

5.1.3 Example problems

In this section, we illustrate how to apply our framework to generic problems in Euclidean space.

[1]Due to space constraints, a convergence proof using the proposed composite Bregman divergence is found in the supplementary materials.

Balanced OT-regularized least squares

We define the balanced OT-regularized inverse problem as

$$\min_{\boldsymbol{x}\in\mathbb{R}^n_+} \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{\Phi}\boldsymbol{x}\|_2^2 + \kappa\mathcal{W}(\boldsymbol{x},\widehat{\boldsymbol{x}}), \tag{5.8}$$

where $\boldsymbol{y}\in\mathbb{R}^m$ denotes measurements, $\boldsymbol{\Phi}\in\mathbb{R}^{m\times n}$ a measurement matrix, $\widehat{\boldsymbol{x}}\in\mathbb{R}^n_+$ a signal prior, and $\mathcal{W}$ a balanced OT-regularizer (i.e., constraints of the form of (BOT)). The first term serves as a data fidelity term while the second is an OT-regularizer that penalizes mass transport away from the prior $\widehat{\boldsymbol{x}}$; both terms are balanced by parameter $\kappa>0$.

First, we expand this to the form of (5.2):

$$\min_{\boldsymbol{x},\boldsymbol{P}} \ \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{\Phi}\boldsymbol{x}\|_2^2 + \iota_+(\boldsymbol{x},\boldsymbol{P}) + \kappa\left\langle \boldsymbol{P},\boldsymbol{C}\right\rangle + \iota_{\boldsymbol{P}^\top\mathbb{1}=\widehat{\boldsymbol{x}}}(\boldsymbol{P}) + \iota_{\boldsymbol{P}\mathbb{1}=\boldsymbol{x}}(\boldsymbol{x},\boldsymbol{P}).$$

Next, we reorganize this into the framework of (5.3) with the following splitting strategy:

$$\begin{aligned}\min_{\boldsymbol{x},\boldsymbol{P},(\boldsymbol{z}_1,\boldsymbol{R}_1),\boldsymbol{z}_2} \quad & f(\boldsymbol{x},\boldsymbol{P}) + \kappa\left\langle \boldsymbol{P},\boldsymbol{C}\right\rangle + \iota_+(\boldsymbol{P}) + g_1(\boldsymbol{z}_1,\boldsymbol{R}_1) + g_2(\boldsymbol{z}_2)\\ \text{s.t.} \quad & \boldsymbol{x}=\boldsymbol{z}_1=\boldsymbol{z}_2, \quad \boldsymbol{P}=\boldsymbol{R}_1,\end{aligned}$$

where

$$\begin{aligned} f(\boldsymbol{x},\boldsymbol{P}) &= \iota_+(\boldsymbol{x}) + \iota_{\boldsymbol{P}^\top\mathbb{1}=\widehat{\boldsymbol{x}}}(\boldsymbol{P}),\\ g_1(\boldsymbol{z}_1,\boldsymbol{R}_1) &= \iota_{\boldsymbol{R}_1\mathbb{1}=\boldsymbol{z}_1}(\boldsymbol{z}_1,\boldsymbol{R}_1),\\ g_2(\boldsymbol{z}_2) &= \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{\Phi}\boldsymbol{z}_2\|_2^2.\end{aligned}$$

Hereon, the update steps are laid out exactly as in (5.5)-(5.7). Since $\boldsymbol{x},\boldsymbol{P}$ are separable, they are solved independently: $\boldsymbol{x}$ is solved by a projection to the non-negative orthant, while $\boldsymbol{P}$ is solved via exponentiated gradients (see [36, Eq. (41)]). To solve $(\boldsymbol{z}_1,\boldsymbol{R}_1)$, we apply the proximal algorithm of Section 5.1.4, and to solve $\boldsymbol{z}_2$, we apply a proximal least-

squares update. Algorithm 4 implements proposed our KL-Euclidean Bregman ADMM method (derivation details can be found in Appendix B.3).

Algorithm 4 KL-Euclidean Bregman ADMM algorithm for the balanced OT-regularized inverse problem.

procedure BALOTINVERSE($\boldsymbol{y}$, $\boldsymbol{\Phi}$, $\boldsymbol{C}$, $\widehat{\boldsymbol{x}}$, κ, ρ, τ)
  $\boldsymbol{x} = \boldsymbol{z}_1 = \boldsymbol{z}_2 = \boldsymbol{\Phi}^\top \boldsymbol{y}$
  $\boldsymbol{P} = \boldsymbol{R}_1 = (\mathbb{1}^\top \widehat{\boldsymbol{x}}/n^2)\mathbb{1}\mathbb{1}^\top$
  $\boldsymbol{a}_1 = \boldsymbol{a}_2 = \boldsymbol{B}_1 = 0$
  while not converged **do**
    $\boldsymbol{x} \leftarrow \max(\frac{1}{2}(\boldsymbol{z}_1 + \boldsymbol{z}_2 - (\boldsymbol{a}_1 + \boldsymbol{a}_2)/\rho), 0)$
    $\boldsymbol{P} \leftarrow \text{PROX1}(\widehat{\boldsymbol{x}}, \kappa\boldsymbol{C}, \boldsymbol{R}, \boldsymbol{B}_1, \rho)$
    $(\boldsymbol{z}_1, \boldsymbol{R}_1) \leftarrow \text{PROX2}(\boldsymbol{x}, \boldsymbol{P}, \boldsymbol{a}_1, \boldsymbol{B}_1, \rho)$
    $\boldsymbol{z}_2 \leftarrow (\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \rho\boldsymbol{I})^{-1}(\boldsymbol{\Phi}^\top\boldsymbol{y} + \rho\boldsymbol{x} + \boldsymbol{a}_2)$
    $\boldsymbol{a}_1 \leftarrow \boldsymbol{a}_1 + \tau(\boldsymbol{x} - \boldsymbol{z}_1)$
    $\boldsymbol{a}_2 \leftarrow \boldsymbol{a}_2 + \tau(\boldsymbol{x} - \boldsymbol{z}_2)$
    $\boldsymbol{B}_1 \leftarrow \boldsymbol{B}_1 + \tau(\boldsymbol{P} - \boldsymbol{R}_1)$
  end while
end procedure

procedure PROX1($\widehat{\boldsymbol{x}}$,$\boldsymbol{C}$,$\boldsymbol{R}$,$\boldsymbol{B}$,ρ) ▷ See [36]
  $\boldsymbol{S} \leftarrow \exp(-(\boldsymbol{C} + \boldsymbol{B})/\rho + \log(\boldsymbol{R}))$
  $\boldsymbol{P} \leftarrow \boldsymbol{S}\,\text{diag}(\widehat{\boldsymbol{x}} \oslash \boldsymbol{S}^\top\mathbb{1})$
end procedure

procedure PROX2($\boldsymbol{x}$,$\boldsymbol{P}$,$\boldsymbol{a}$,$\boldsymbol{B}$,ρ) ▷ See Section 5.1.4
  $\boldsymbol{S} \leftarrow \exp(\boldsymbol{B}/\rho + \log(\boldsymbol{P}) + (\boldsymbol{a}/\rho + \boldsymbol{x})\mathbb{1}^\top)$
  $\boldsymbol{z} \leftarrow \omega(\boldsymbol{S}\mathbb{1})$
  $\boldsymbol{R} \leftarrow \text{diag}(\mathbb{1} \oslash \exp(\boldsymbol{z}))\boldsymbol{S}$
end procedure

Unbalanced problems

A naïve approach of splitting the inverse problem with unbalanced OT-regularizers (UOT), is to split the problem for each constraint. This however generates redundant variables that reduce the rate of convergence and utilize more memory. Instead we exploit a redundancy in the constraints with slack variables; specifically, we introduce variables $\boldsymbol{s}$ and $\widehat{\boldsymbol{s}}$ to respectively represent non-negative growth and destruction using the decomposition

$(\boldsymbol{x}, \widehat{\boldsymbol{x}}, \boldsymbol{s}, \widehat{\boldsymbol{s}})$ in place of $(\boldsymbol{z}, \widehat{\boldsymbol{z}})$. The function H_μ in (UOT) may be reexpressed:

$$H_\mu((\boldsymbol{x}, \widehat{\boldsymbol{x}}, \boldsymbol{s}, \widehat{\boldsymbol{s}}), \boldsymbol{P}) = \mu(\|\boldsymbol{s}\|_p^p + \|\widehat{\boldsymbol{s}}\|_p^p) + \iota_+(\boldsymbol{x}, \widehat{\boldsymbol{x}}, \boldsymbol{s}, \widehat{\boldsymbol{s}}) \\ + \iota_{\boldsymbol{P}\mathbb{1}=\boldsymbol{x}+\boldsymbol{s}}(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{P}) + \iota_{\boldsymbol{P}^\top\mathbb{1}=\widehat{\boldsymbol{x}}+\widehat{\boldsymbol{s}}}(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{s}}, \boldsymbol{P}),$$

and cast into the framework of (5.3) with auxiliary variables $\boldsymbol{z}, \boldsymbol{R}$ as

$$\begin{aligned} \min_{(\boldsymbol{x}, \widehat{\boldsymbol{x}}, \boldsymbol{s}, \widehat{\boldsymbol{s}}), \boldsymbol{P}, (\boldsymbol{z}, \boldsymbol{R})} \quad & f_0(\boldsymbol{x}, \widehat{\boldsymbol{x}}) + \iota_+(\boldsymbol{P}) + \kappa \langle \boldsymbol{P}, \boldsymbol{C} \rangle + f_1(\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{P}) + g(\widehat{\boldsymbol{s}}, \boldsymbol{z}, \boldsymbol{R}) \\ \text{s.t.} \quad & \widehat{\boldsymbol{x}} = \boldsymbol{z}, \quad \boldsymbol{P} = \boldsymbol{R}, \end{aligned} \tag{5.9}$$

where f_0 is a generic convex loss function, and

$$f_1(\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{P}) = \mu \|\boldsymbol{s}\|_p^p + \iota_+(\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{P}) + \iota_{\boldsymbol{P}\mathbb{1}=\boldsymbol{x}+\boldsymbol{s}}(\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{P}),$$
$$g(\widehat{\boldsymbol{s}}, \boldsymbol{z}, \boldsymbol{R}) = \mu \|\widehat{\boldsymbol{s}}\|_p^p + \iota_+(\widehat{\boldsymbol{s}}, \boldsymbol{z}, \boldsymbol{R}) + \iota_{\boldsymbol{R}^\top\mathbb{1}=\boldsymbol{z}+\widehat{\boldsymbol{s}}}(\widehat{\boldsymbol{s}}, \boldsymbol{z}, \boldsymbol{R}).$$

In the next section, we provide proximal algorithms that admit closed form solutions to the above problems for $p = 1$ and $p = 2$.

5.1.4 KL-Euclidean Bregman proximal algorithms

In this section, we demonstrate efficient proximal algorithms that solve ADMM updates (5.5) and (5.6), which contain both KL and quadratic ℓ_2 proximal terms. We point out two broad categories of constraints that are described with H (defined in Section 5.1.1): those dependent only on $\boldsymbol{R}$ and those jointly dependent on $\boldsymbol{z}$ and $\boldsymbol{R}$. In the first category, numerous proximal algorithms exist based on exponentiated gradients [36, 141, 65]. When there is joint dependency on both $(\boldsymbol{z}, \boldsymbol{R})$, we show that closed form proximal algorithms still result. Due to space limitations, we only present proximal algorithms for three commonly used marginal constraints (balanced, unbalanced with $p = 1$, and unbalanced with $p = 2$). Proximal algorithms for other types of constraints (e.g., marginal-inequalities, capacity,

see [65] for examples) can be easily derived using similar proof techniques as those used in Appendix B.2. We now proceed with solving the following proximal problems under the assumption of feasibility and existence of an optimal solution; we note that the Bregman ADMM sequence guarantees feasibility when the step-size conditions are satisfied.

Balanced marginal constraints

We have the problem

$$\min_{\boldsymbol{R}\mathbb{1}=\boldsymbol{z}} \ \langle \boldsymbol{a}, -\boldsymbol{z}\rangle + \langle \boldsymbol{B}, -\boldsymbol{R}\rangle + \rho B(\boldsymbol{z}, \boldsymbol{R}|\boldsymbol{x}, \boldsymbol{P}), \tag{5.10}$$

where B is defined in (5.4).

Theorem 5.1.3. *The problem* (5.10) *has optimal solution* $(\boldsymbol{z}^\star, \boldsymbol{R}^\star)$ *given in closed form by*

$$\boldsymbol{z}^\star = \omega(\boldsymbol{S}\mathbb{1}), \quad \boldsymbol{R}^\star = \operatorname{diag}(1/\exp(\boldsymbol{z}^\star))\boldsymbol{S},$$

where $\boldsymbol{S} = \exp((\boldsymbol{a}/\rho + \boldsymbol{x})\mathbb{1}^\top + \boldsymbol{B}/\rho + \log \boldsymbol{P})$ *and* $\omega(\cdot)$ *is the element-wise Lambert W function.*

Proof. See Appendix B.2.1.

Unbalanced marginal constraint, $p = 2$

We have the problem

$$\begin{aligned}\min_{\boldsymbol{R}\mathbb{1}=\boldsymbol{z}+\boldsymbol{u}} \ & \iota_+(\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{R}) + \frac{\mu}{2}\|\boldsymbol{u}\|_2^2 + \langle \boldsymbol{a}, -\boldsymbol{z}\rangle + \langle \boldsymbol{B}, -\boldsymbol{R}\rangle \\ & + \rho B(\boldsymbol{z}, \boldsymbol{R}|\boldsymbol{x}, \boldsymbol{P}),\end{aligned} \tag{5.11}$$

where B is defined in (5.4).

Theorem 5.1.4. *The problem* (5.11) *has optimal solution* $(\boldsymbol{z}^\star, \boldsymbol{R}^\star)$ *given in closed form by*

$$z^\star_{i\in\Omega_k} = z_i^{(k)}, \quad R^\star_{i\in\Omega_k,:} = R^{(k)}_{i,:}, \quad k \in \{1,2,3\},$$

where the support sets are defined as

$$\begin{aligned}
\Omega_1 &= \{i \in [\![n]\!] : (\boldsymbol{R}^{(1)}\mathbb{1})_i > z_i^{(1)} > 0\},\\
\Omega_2 &= \{i \in [\![n]\!] : (\boldsymbol{R}^{(1)}\mathbb{1})_i \le z_i^{(1)} > 0\},\\
\Omega_3 &= \{i \in [\![n]\!] : z_i^{(1)} \le 0\},
\end{aligned}$$

and solutions related to each set are defined as

$$\begin{aligned}
\boldsymbol{z}^{(1)} &= \omega(((\boldsymbol{S}\mathbb{1}) \odot \exp(-\boldsymbol{p}/\alpha))/\alpha) + \boldsymbol{p}/\alpha,\\
\boldsymbol{R}^{(1)} &= \operatorname{diag}(\exp(-\boldsymbol{z}^{(1)}))\boldsymbol{S},\\
\boldsymbol{z}^{(2)} &= \omega(\boldsymbol{S}\mathbb{1}),\\
\boldsymbol{R}^{(2)} &= \operatorname{diag}(\exp(-\boldsymbol{z}^{(2)}))\boldsymbol{S},\\
\boldsymbol{z}^{(3)} &= 0,\\
\boldsymbol{R}^{(3)} &= \operatorname{diag}(\exp(-\omega(\beta\boldsymbol{Q}\mathbb{1})/\beta))\boldsymbol{Q},
\end{aligned}$$

where $\boldsymbol{S} = \exp((\boldsymbol{a}/\rho + \boldsymbol{x})\mathbb{1}^\top + \boldsymbol{B}/\rho + \log \boldsymbol{P})$, $\boldsymbol{p} = (\boldsymbol{a} + \rho\boldsymbol{x})/\mu$, $\alpha = 1 + \rho/\mu$, $\boldsymbol{Q} = \exp(\log \boldsymbol{P} + \boldsymbol{B}/\rho)$, $\beta = \mu/\rho$, *and* $\omega(\cdot)$ *is the element-wise Lambert W function.*

Proof. See Appendix B.2.2.

Unbalanced marginal constraint, $p = 1$

We have the problem

$$\begin{aligned}
\min_{\boldsymbol{R}\mathbb{1}=\boldsymbol{z}+\boldsymbol{u}} \; & \iota_+(\boldsymbol{u},\boldsymbol{z},\boldsymbol{R}) + \mu\left\|\boldsymbol{u}\right\|_1 + \langle \boldsymbol{a}, -\boldsymbol{z}\rangle + \langle \boldsymbol{B}, -\boldsymbol{R}\rangle\\
& + \rho B(\boldsymbol{z},\boldsymbol{R}|\boldsymbol{x},\boldsymbol{P}),
\end{aligned} \tag{5.12}$$

where B is defined in (5.4).

Theorem 5.1.5. *The problem* (5.12) *has optimal solution* $(\boldsymbol{z}^\star, \boldsymbol{R}^\star)$ *given by*

$$z^\star_{i\in\Omega_k} = z^{(k)}_i, \quad R^\star_{i\in\Omega_k,:} = R^{(k)}_{i,:}, \quad k \in \{1,2\},$$

where the support sets are defined as

$$\Omega_1 = \{i \in [\![n]\!] : (\boldsymbol{R}^{(1)}\mathbb{1})_i > z^{(1)}_i\},$$
$$\Omega_2 = \{i \in [\![n]\!] : (\boldsymbol{R}^{(1)}\mathbb{1})_i \leq z^{(1)}_i\},$$

and solutions related to each set are defined as

$$\boldsymbol{z}^{(1)} = (\mu\mathbb{1} + \boldsymbol{a})/\rho + \boldsymbol{x}, \quad \boldsymbol{R}^{(1)} = \exp(\log \boldsymbol{P} + \boldsymbol{B}/\rho - \mu/\rho),$$
$$\boldsymbol{z}^{(2)} = \omega(\boldsymbol{S}\mathbb{1}), \qquad\qquad \boldsymbol{R}^{(2)} = \operatorname{diag}(1/\exp(\boldsymbol{z}^{(2)}))\boldsymbol{S},$$

where $\boldsymbol{S} = \exp(\log \boldsymbol{P} + \boldsymbol{B}/\rho + (\boldsymbol{a}/\rho + \boldsymbol{s})\mathbb{1}^\top)$*, and* $\omega(\cdot)$ *is the element-wise Lambert W function.*

Proof. See Appendix B.2.3.

5.2 Performance of KL-Euclidean Bregman ADMM

In this section, we test the performance of the proposed solver on the balanced OT-regularized inverse problem (introduced in Section 5.1.3). We adopt a similar style of experiments as [142, §4] and use a compressive sensing matrix $\boldsymbol{\Phi} \in \mathbb{R}^{m\times n}$ $(m < n)$ with i.i.d. random Gaussian entries.

5.2.1 Comparison of ADMM methods

We compare the performance of the proposed KL-Euclidean *Bregman ADMM* solver against methods that use the standard ADMM framework [22] (i.e., quadratic ℓ_2 proximity dis-

tance), such as (i) simplex projections [163] (which we call *Euclidean ADMM*), and (ii) generalized Sinkhorn iterations [46] (which we call *Sinkhorn ADMM*). Although various first-order proximal algorithms exist, all compared methods are solved using an ADMM framework for consistency.

We note several differences between the methods. First, the Bregman ADMM and Euclidean ADMM solve the inverse problem with OT-regularization precisely, while Sinkhorn ADMM solves only an approximation of the OT-regularizer (via entropic regularization). We tune the Sinkhorn approximation parameter to achieve maximum precision within the limits of numerical stability ($\epsilon = 10^{-1}$ in Algorithm 1 of [46]). Second, compared to other methods, Sinkhorn ADMM does not explicitly access the transport coupling matrix $\boldsymbol{P}$; rather, it does so implicitly via an iterative Sinkhorn proximal algorithm. In practice, we found the sizable number of Sinkhorn iterations required for convergence to be a computational bottleneck. To level the playing field between methods, we artificially limit the iterative Sinkhorn proximal algorithm to only a *single* iteration and provide it with warm start capabilities. Derivation details for these methods are found in Appendix B.3. In Figure 5.1, we plot three metrics over the course of the ADMM iterations: relative error (i.e., difference between $\boldsymbol{x}^{(t)}$ and the ground truth), primal residual norm (i.e., norm of difference between splitting variables and the primal), and dual residual norm (i.e., norm of difference in splitting variables from consecutive ADMM iterations). Bregman ADMM and Sinkhorn ADMM are observed to have a comparable convergence rate that is superior to Euclidean ADMM's. This is consistent with the $O(n/\log n)$ improvement factor shown by [36, 35], since the KL-divergence proximity distance allows proximal gradient descent to be optimal over the simplex set.

Compared to Bregman ADMM, Sinkhorn ADMM only solves the OT-regularizer approximately with an accuracy inversely proportional to the amount of entropic regularization. We therefore advocate our approach in applications that demand accuracy. Despite the comparable convergence rate of Bregman ADMM and Sinkhorn ADMM, we note

that Sinkhorn ADMM has roughly half the per-iteration computational cost of Bregman ADMM ($O(4n^2)$ versus $O(9n^2)$). This is expected because Sinkhorn ADMM has very efficient machinery (i.e., block coordinate dual ascent) specifically configured for balanced OT constraints (BOT); Bregman ADMM instead has a very different structural setup (primal descent and dual ascent) that affords a flexible mechanism easily adaptable to various types of OT variants such as (BOT), (UOT) as well as others (e.g., [65]). On the other hand, reconfiguring [46] for another OT setup requires non-trivial rederivations. Moreover, differences in computational efficiency becomes negligible when the algorithm is decentralized and parallelized. We concede that the increased versatility of our method comes at the cost of increased memory consumption (required to store auxiliary and dual variables); for example, our solver approach to problem (5.8) requires $4\times$ more memory than [46].

5.2.2 Comparison against a benchmark second order solver

Again using the balanced OT-regularized inverse problem of the previous subsection, we compare our proposed method against a second-order interior point solver in their respective optimized implementations. This problem is a linearly constrained quadratic program, which interior point solvers are known to efficiently solve to high accuracy. Although first order methods do not solve problems to as high an accuracy as second order methods, we ensure that our solver produces solutions of sufficient accuracy before termination. Specifically, by using the interior point method's solution as an accuracy benchmark, we find that an ADMM tolerance of 10^{-2} produces comparably accurate solutions. Figure 5.2 illustrates how various termination tolerances (defined as the value which the primal and dual residual norms must both satisfy) effects the solution's accuracy at various compression ratios m/n. We measure accuracy via two metrics, relative error (for reconstruction accuracy) and F1 score (for support estimation).

The proposed KL-Euclidean Bregman ADMM solver is optimally implemented on a general-purpose graphics processing unit (GPGPU) since the dominant costs are due to

Table 5.1: Comparison in speed and accuracy between our method and commercial interior point solver Gurobi [62] on problem (5.8)

Problem size (n)	Gurobi runtime (s)	BADMM runtime (s)	BADMM Speedup	Rel. obj. diff. (10^{-5})
16^2	0.15 ± 0.04	0.042 ± 0.01	$3.7\times$	29.9
32^2	2.14 ± 0.06	0.237 ± 0.03	$9.0\times$	6.90
64^2	98.2 ± 3.26	4.293 ± 0.46	$22.9\times$	3.38
96^2	959 ± 31.9	26.31 ± 2.84	$36.5\times$	3.32
128^2	(Out of memory)	94.12 ± 8.82	-	-

matrix operations that can be efficiently computed on parallelized hardware. Specifically, we used an Nvidia GeForce GTX 1080 GPGPU which has 2560 cores clocked at 1.733 GHz and 8 GB of RAM. We note that interior point solvers are optimally implemented on CPU architecture (rather than on GPGPUs) since the main bottleneck per interior point iteration is a dense Cholesky factorization which by nature is serial[2] [168]. Specifically, this was implemented with the highly optimized commercial solver Gurobi [62] using a 28 core Intel(R) Xeon(R) CPU (E5-2680 v4) clocked at 2.40 GHz, with 128 GB of memory.

We compare the solvers' run times and (relative) differences between their objectives on various problem sizes, $n = \{16^2, 32^2, 64^2, 96^2, 128^2\}$. For each problem size, 10 random synthetic trials were performed and their mean and standard deviation reported in Table 5.1. The results demonstrate superior efficiency of our solver at all problem sizes, with a maximum speedup measured at $37\times$ for the problem size of $n = 96^2$. BADMM's objective value (o_b) is also found to be similar to that of Gurobi's (o_g) using the *relative objective difference* metric defined as $|o_b - o_g|/o_g$. Lastly, our method also exhibits improved memory efficiency: on the largest problem ($n = 128^2$), Gurobi runs out 128 GB of memory (on CPU), while our solver is still capable of solving the problem on a significantly limited 8 GB of memory (on GPGPU).

[2]Scalable Cholesky factorization performance is still an area of open research on parallel-computing platforms, with highly-customized hardware configurations currently showing the most promise [164, 165, 166, 167].

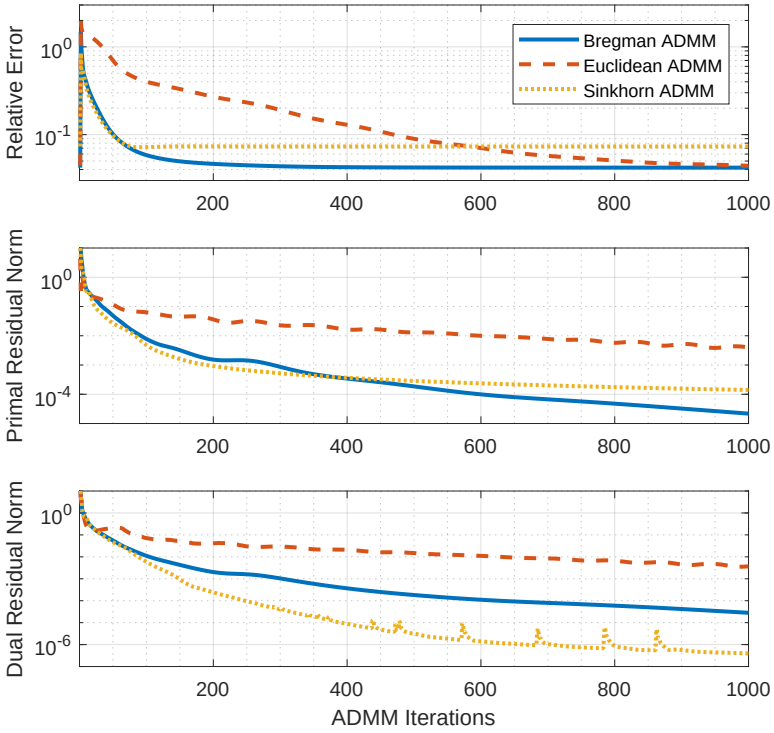


Figure 5.1: Convergence rates of ADMM methods. Our proposed method (labeled Bregman ADMM) has a comparable rate of convergence as a fast proximal method in recent computational optimal transport literature [46] (labeled Sinkhorn ADMM). Both of these methods converge significantly faster than standard ADMM [22] (labeled Euclidean ADMM). Unlike Sinkhorn ADMM, our proposed method does not approximate the OT-regularizer and therefore produces more accurate reconstructions (i.e., lower relative error at convergence).

5.3 Applications

In this section, we demonstrate the versatility of OT-regularization on two novel applications that have become tractable with our framework.

5.3.1 Connectome-informed EEG tracking with OT-regularized BPDN

In this section, we apply an OT-regularizer to the tracking of time-varying electroencephalography (EEG) sources. The goal of EEG source localization is to spatially pin-point active sources of electrical activation $\boldsymbol{x}$ within the brain from coarse scalp measurements $\boldsymbol{y}$ using

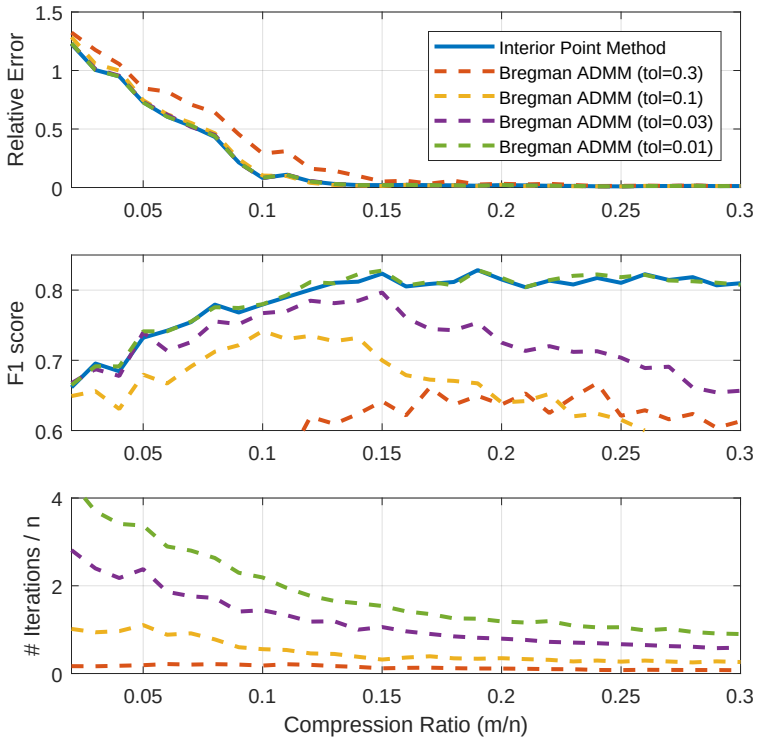


Figure 5.2: Reconstruction accuracy versus termination tolerance. Based on two metrics of accuracy, relative error (for reconstruction accuracy) and F1 score (for support estimation), we observe that a tolerance of 10^{-2} yields solutions with accuracies that are comparable to the highly accurate solution produced by an interior point method. The third plot describes the required number of iterations (scaled by the dimensions n) to achieve the prescribed tolerance.

a forward linear model of the electromagnetic field $\mathbf{\Phi}$ obtained from Maxwell's equations. In general, this is a challenging problem since (i) the problem is highly underdetermined, and (ii) the measurement matrix $\mathbf{\Phi}$ has highly correlated entries. We note that a complete review of the EEG source localization literature is beyond the scope of this paper, although we do point to two foundational methods that employ structure to improve the conditioning of the inverse problem: [169] uses Tikhonov-regularization, and [170] exploits sparsity via the LASSO [3]. Since brain states are known to have temporally-correlated activity related to functional network-connectivity within the brain [171, 172], we propose integrating dynamical priors with structural information to further improve inference. To that end, we use the OT mechanism as the way of fusing functional network-connectivity information

with dynamical information.

We consider the real-valued variant of the earth-mover's distance dynamical filtering (EMD-DF) problem [143] which is formulated as

$$\widehat{\boldsymbol{x}}_k = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\arg\min}\ \frac{1}{2}\|\boldsymbol{y}_k - \boldsymbol{\Phi}_k\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 + \kappa\mathcal{W}(|\boldsymbol{x}|, |\widehat{\boldsymbol{x}}_{k-1}|). \tag{5.13}$$

where k refers to a time index and $\mathcal{W}$ may be either a partial or unbalanced transport variant. This method was shown in [143] to be amenable to applications like frequency tracking (in the complex domain), where signal magnitude (sans phase) is transported across frequency bins. We describe a real-valued model that weighs positive and negative signals similarly as mass, i.e., $\mathcal{F}$ in $\mathcal{W}(\mathcal{F}(\cdot), \mathcal{F}(\cdot))$ is an absolute operator (i.e., $|\cdot|$). Following [143], we make this convex using a positive-negative decomposition:

$$\begin{aligned}\underset{\boldsymbol{x}^+,\boldsymbol{x}^-\in\mathbb{R}_+^n}{\arg\min}\ & \frac{1}{2}\|\boldsymbol{y}_k - \boldsymbol{\Phi}_k(\boldsymbol{x}^+ - \boldsymbol{x}^-)\|_2^2 + \lambda\|\boldsymbol{x}^+ + \boldsymbol{x}^-\|_1 \\ & + \kappa\mathcal{W}(\boldsymbol{x}^+ + \boldsymbol{x}^-, |\widehat{\boldsymbol{x}}_{k-1}|).\end{aligned}$$

where $|\widehat{\boldsymbol{x}}_{k-1}| = \widehat{\boldsymbol{x}}_{k-1}^+ + \widehat{\boldsymbol{x}}_{k-1}^-$ refers to the signal estimate from the previous frame. We apply the unbalanced OT-regularizer (UOT), and solve the resulting problem with variables $\boldsymbol{s}$ and $\widehat{\boldsymbol{s}}$ which represent mass growth and decay, respectively. This is expanded to the form of (5.2) as

$$\begin{aligned}\underset{(\boldsymbol{x}^+,\boldsymbol{x}^-,\boldsymbol{s},\widehat{\boldsymbol{s}},\boldsymbol{w}),\boldsymbol{P}}{\arg\min}\quad & f_0(\boldsymbol{x}^+,\boldsymbol{x}^-,\boldsymbol{s},\widehat{\boldsymbol{s}}) + \iota_+(\boldsymbol{P}) + \kappa\left\langle \boldsymbol{P}, \boldsymbol{C}\right\rangle \\ & + \iota_{\boldsymbol{P}\mathbb{1}=\boldsymbol{w}}(\boldsymbol{w},\boldsymbol{P}) + \iota_{\boldsymbol{P}^\top\mathbb{1}=|\widehat{\boldsymbol{x}}_{k-1}|+\widehat{\boldsymbol{s}}}(\widehat{\boldsymbol{s}},\boldsymbol{P}) \\ \text{s.t.}\quad & \boldsymbol{w} = \boldsymbol{x}^+ + \boldsymbol{x}^- + \boldsymbol{s},\end{aligned}$$

where

$$\begin{aligned}f_0(\boldsymbol{x}^+,\boldsymbol{x}^-,\boldsymbol{s},\widehat{\boldsymbol{s}}) = \lambda\left\|\boldsymbol{x}^+ + \boldsymbol{x}^-\right\|_1 + \mu(\|\boldsymbol{s}\|_p^p + \|\widehat{\boldsymbol{s}}\|_p^p) \\ +\iota_+(\boldsymbol{x}^+,\boldsymbol{x}^-,\boldsymbol{s},\widehat{\boldsymbol{s}}).\end{aligned}$$

Finally, we reorganize this into the splitting framework of (5.3) with auxiliary variables $\boldsymbol{z}_1, \boldsymbol{z}_2$, and $\boldsymbol{R}$:

$$
\begin{aligned}
&f_1(\boldsymbol{x}_1, \boldsymbol{P}) = \iota_{\boldsymbol{P}\mathbb{1}=\boldsymbol{w}}(\boldsymbol{w}, \boldsymbol{P}) \\
&g_1(\boldsymbol{z}_1, \boldsymbol{R}) = \iota_{\boldsymbol{R}^\top \mathbb{1}=\boldsymbol{z}_1}(\boldsymbol{z}_1, \boldsymbol{R}), \quad \boldsymbol{z}_1 = |\widehat{\boldsymbol{x}}_{k-1}| + \widehat{\boldsymbol{s}}, \quad \boldsymbol{P} = \boldsymbol{R}, \\
&g_2(\boldsymbol{z}_2) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{z}_2\|_2^2, \qquad\quad \boldsymbol{z}_2 = \boldsymbol{x}^+ - \boldsymbol{x}^-.
\end{aligned}
$$

This program allows the generic update steps described by (5.5)-(5.7). We exploit the separable structure of $f(\boldsymbol{x}^+, \boldsymbol{x}^-, \boldsymbol{s}, \widehat{\boldsymbol{s}})$ and $g_2(\boldsymbol{z}_2)$ and solve them with standard Euclidean proximal methods [22], while $f_1(\boldsymbol{w}, \boldsymbol{P})$ and $g_1(\boldsymbol{z}_1, \boldsymbol{R}_1)$ require the balanced marginal-equality KL-Euclidean Bregman proximal algorithm described in Section 5.1.4.

In the final stages of preparing this manuscript, we became aware of a similar concurrent development in OT-regularization for source-localization in EEG applications [173]. This work utilizes an unbalanced OT-regularizer [61] in the inverse EEG problem, albeit involving a KL divergence for the marginal relaxation. Numerically, [173] utilizes a Sinkhorn distance which, as demonstrated in Section 5.2.1, trades reconstruction fidelity for speed; our method does not suffer from this trade-off. Additionally, model wise, we consider a single-subject over time and describe ground-costs with brain-interconnections; [173] considers multi-subject snapshots, and therefore describe ground-costs by physical geodesic curvature along the cortex surface.

In our simulations, the spatial coordinates of EEG dipole sources $\boldsymbol{x}_k \in \mathbb{R}^n$ ($n = 1261$) were specified by sampling a skull volume on a uniform 3D grid. Scalp measurements $\boldsymbol{y}_k \in \mathbb{R}^m$ ($m = 256$) were generated as $\boldsymbol{y}_k = \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\eta}_k$ where $\boldsymbol{\Phi}_k \in \mathbb{R}^{m \times n}$ is an electromagnetic model of Maxwell's equations (generated using the boundary element method [174]), along with additive Gaussian noise that models sensor noise $\boldsymbol{\eta}_k \sim \mathcal{N}(0, \sigma \boldsymbol{I}_m)$ (with $\sigma = 10^{-5}$). We let the source dipoles inherit interconnections $\boldsymbol{F} = \{0, 1\}^{n \times n}$ based on a brain connectivity network known as the *default mode network* (DMN) [171] via spatial registration to anatomical brain parcellations. Specifically, we define $F_{ij} = 1$ with proba-

bility $1-e^{-1/0.2}$ if i, j are within the DMN; conversely $F_{ij} = 0$ with probability $1-e^{-1/0.01}$ if i or j are outside the DMN. For a physical visualization, see the top left image of Figure 5.3. We describe the propagation of source activations as masses "hopping" through a network at intervals of τ, where the probability of propagation $p(\boldsymbol{n}_i \to \boldsymbol{n}_j)$ is $1/|\mathcal{N}_i|$ if $\boldsymbol{n}_j \in \mathcal{N}_i$, where $\mathcal{N}_i$ is the set of neighbors connected to node $\boldsymbol{n}_i$, and 0 otherwise. In the period between the l-th and $(l+1)$-th hop, a half-sine wave pulse is generated at the i-th node as $x_{(l-1)\tau+t}(i) = sin(t\pi/(\tau+1))$ for $t = 1, \ldots, \tau$. We show source voxels activations over time as dotted black lines in the top right plot of Figure 5.3. For reconstruction, we assume $\boldsymbol{F}$ is known and construct the OT ground cost matrix to be inversely proportional to it, i.e., $C_{ij} = 1/(F_{ij} + \eta)$, where $\eta > 0$ is a small constant (e.g., 10^{-2}).

In Figure 5.3, we demonstrate superior performance in both reconstruction error, as measured with *relative mean square error* (rMSE), and localization performance, as measured using the F1-score. rMSE is computed as $\|\widehat{\boldsymbol{x}} - \boldsymbol{x}^\star\|_2^2 \,/\, \|\boldsymbol{x}^\star\|_2^2$, where $\boldsymbol{x}^\star$ is ground truth and $\widehat{\boldsymbol{x}}$ is the approximation. F1-score is computed as $2(\|\boldsymbol{m}^\star \cap \widehat{\boldsymbol{m}}\|_0)/(\|\boldsymbol{m}^\star\|_0 + \|\widehat{\boldsymbol{m}}\|_0)$, where $m_i = 1$ if $x_i \geq$ threshold and 0 otherwise, and $\|\cdot\|_0$ denotes the cardinality of its argument. The range of the rMSE is $[0, \infty]$ with 0 indicating perfect reconstruction, while the range of the F1-score is $[0, 1]$ with 1 indicating perfect localization. We compared our proposed algorithm (5.13) against two classical approaches: minimum-norm estimation (MNE) [169] (i.e., Tikhonov-regularized least-squares), and minimum current estimation (MCE) [170] (i.e., the LASSO [3]). We plot the mean performance for both metrics over 40 random time series, with error bars denoting ± 1 standard deviation.

5.3.2 Interference-source separation for audio chirps with OT-regularized RPCA

In this section, we develop a sparse outlier tracking method based on the celebrated robust principal components analysis (RPCA) framework, a versatile subspace/sparse-outlier separation tool used in applications such as facial recognition and video surveillance [139]. RPCA extends the regular principal components analysis framework by modeling data $\boldsymbol{X}$

as a superposition of low-rank subspaces $\boldsymbol{L}$ and sparsely – rather than Normally – distributed outliers $\boldsymbol{S}$, improving robustness to gross outliers. Using this model, accurate recovery of $\boldsymbol{L}$ and $\boldsymbol{S}$ may be guaranteed with high probability under incoherence and randomness conditions (see [139, 151]) by solving the convex program $\min_{\boldsymbol{S},\boldsymbol{L}} \ \lambda\|\boldsymbol{S}\|_1+\|\boldsymbol{L}\|_*$ s.t. $\boldsymbol{X} = \boldsymbol{S} + \boldsymbol{L}$. We utilize a particular variant of RPCA [155] which performs separation with an additional Gaussian noise term $\boldsymbol{N}$ in the superposition. We extend RPCA to scenarios that involve tracking sparse components within an ordered dataset using the first order Markovian model $\boldsymbol{s}_{t+1} = f(\boldsymbol{s}_t)$, where $\boldsymbol{s}_t \in \mathbb{R}^n_+$ denotes the t-th column in $\boldsymbol{S}$ and f encapsulates correlations between adjacent columns in $\boldsymbol{S}$ under mild mass-preservation constraints. This model has potential applications ranging from continuous line separation in images (e.g., Figure 10 of [175] or [176]) to drum-vocal separation in time-frequency data [177].

Here, we demonstrate our method with time-frequency tracking data. Separating frequency-modulated signals from interference and noise is an important preprocessing step that has been shown to increase the signal-to-interference-and-noise ratio (SINR) for accurate downstream feature analysis such as classification/clustering tasks (e.g., mammalian auditory studies [179, 180] and radar applications [181]). In this work, we apply our method as a novel way to isolate low-energy dolphin whistles (i.e., chirps) from significant background interference-and-noise. Figure 5.4 shows examples of such problems from [178] with varying SINR difficulty. Here, we consider the extended mixture model $\boldsymbol{X} = \boldsymbol{S} + \boldsymbol{N} + \boldsymbol{L}$, where $\boldsymbol{X}$ refers to the STFT magnitude of an audio recording, $\boldsymbol{S}$ represents sparse linear combinations of frequency-modulated signals (chirps), $\boldsymbol{N}$ accounts for white Gaussian-distributed background noise, and $\boldsymbol{L}$ collects low rank structured background interference (e.g., low-frequency engine hum). Here, $\boldsymbol{X}, \boldsymbol{S}, \boldsymbol{N}, \boldsymbol{L} \in \mathbb{R}^{n\times T}_+$ are understood as matrices with columns denoting individual $\mathbb{R}^n$ datapoints in time. The strong presence of sparse specra-wide instantaneous saturations (due to echolocation clicks and buoy motion) and low-frequency engine "sputters" makes the sparsity-encouraging ℓ_1 prior in RPCA ineffec-

tive at isolating chirps. We therefore introduce OT-regularization to add spatial-temporal structure on $\boldsymbol{S}$; when f evolves activations smoothly along the spatial support of $\boldsymbol{s}_t$, the OT-regularizer serves to captures inter-support transitions under mass-preservation constraints. Further, when f has piecewise smooth structure in time, $\mathcal{W}$ may be modeled by the unbalanced OT-regularizer (UOT) with $p = 1$. The resulting optimization procedure combines RPCA with OT-regularization as:

$$\begin{aligned} \min_{\boldsymbol{S},\boldsymbol{N},\boldsymbol{L}} \ & \frac{\beta}{2}\|\boldsymbol{N}\|_F^2 + \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 + \kappa\sum_{i=1}^{T-1}\mathcal{W}_\mu(\boldsymbol{s}_i, \boldsymbol{s}_{i+1}) \\ \text{s.t.} \quad & \boldsymbol{X} = \boldsymbol{S} + \boldsymbol{N} + \boldsymbol{L}, \quad \boldsymbol{S}, \boldsymbol{N} \geq 0. \end{aligned} \tag{5.14}$$

This problem is numerically more challenging than previous work in two ways. First, we have an OT-regularizer that has variability in both arguments (rather than only one). Second, we simultaneously solve a large number ($T - 1$) of OT-regularizers within a single program. The numerical examples shown in Figure 5.4 solve problems containing 800 to 900 OT-regularizers. Our proposed KL-Bregman ADMM framework tackles both these challenges. The first challenge is tackled by the splitting of marginal constraints with our KL-Euclidean Bregman method, while the second is handled via consensus ADMM in distributed implementations (e.g., using tools like MapReduce [182], a popular programming model for distributed batch processing of very large datasets).

To ease notation in the following, we denote columns of matrices by their lower-case letters. To solve this problem with our proposed framework using the unbalanced problem (5.9) as our prototype, we expand it to the form of (5.2):

$$\begin{aligned} \min_{(\boldsymbol{S},\boldsymbol{N},\boldsymbol{L},\boldsymbol{U},\boldsymbol{V}),\{\boldsymbol{P}_t\}_t} f_0(\boldsymbol{S},\boldsymbol{N},\boldsymbol{L}) + \sum_{i=1}^{T-1}\Big\{ & \langle \kappa\boldsymbol{C}, \boldsymbol{P}_t\rangle + \iota_+(\boldsymbol{P}_t) \\ & + f_t(\boldsymbol{u}_t, \boldsymbol{s}_t, \boldsymbol{P}_t) + g_{t+1}(\boldsymbol{v}_{t+1}, \boldsymbol{s}_{t+1}, \boldsymbol{P}_t)\Big\}, \end{aligned}$$

where

$$f_0(\boldsymbol{S}, \boldsymbol{N}, \boldsymbol{L}) = \frac{\beta}{2} \|\boldsymbol{N}\|_F^2 + \lambda \|\boldsymbol{S}\|_1 + \|\boldsymbol{L}\|_* + \iota_+(\boldsymbol{S}, \boldsymbol{N}),$$
$$f_t(\boldsymbol{u}, \boldsymbol{s}, \boldsymbol{P}) = \iota_+(\boldsymbol{u}, \boldsymbol{s}) + \iota_{\boldsymbol{P}\mathbb{1}=\boldsymbol{s}+\boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{s}, \boldsymbol{P}) + \mu \|\boldsymbol{u}\|_p^p,$$
$$g_t(\boldsymbol{v}, \boldsymbol{s}, \boldsymbol{P}) = \iota_+(\boldsymbol{v}, \boldsymbol{s}) + \iota_{\boldsymbol{P}^\top\mathbb{1}=\boldsymbol{v}+\boldsymbol{s}}(\boldsymbol{v}, \boldsymbol{s}, \boldsymbol{P}) + \mu \|\boldsymbol{v}\|_p^p.$$

Reorganize this into the framework of (5.3) with the splitting variables $\boldsymbol{W}, \boldsymbol{Z}, \{\boldsymbol{R}_t\}_t$:

$$\begin{aligned}\min_{\substack{(\boldsymbol{S},\boldsymbol{N},\boldsymbol{L},\boldsymbol{U},\boldsymbol{V}),\{\boldsymbol{P}_t\}_t \\ (\boldsymbol{W},\boldsymbol{Z}),\{\boldsymbol{R}_t\}_t}} & f_0(\boldsymbol{S}, \boldsymbol{N}, \boldsymbol{L}) + \sum_{i=1}^{T-1} \Big\{ \langle \kappa \boldsymbol{C}, \boldsymbol{P}_t \rangle + \iota_+(\boldsymbol{P}_t) \\ & \qquad + f_t(\boldsymbol{u}_t, \boldsymbol{w}_t, \boldsymbol{P}_t) + g_{t+1}(\boldsymbol{v}_{t+1}, \boldsymbol{z}_{t+1}, \boldsymbol{R}_t) \Big\}, \\ \text{s.t.} \quad & \boldsymbol{X} = \boldsymbol{S} + \boldsymbol{N} + \boldsymbol{L}, \quad \boldsymbol{S} = \boldsymbol{W} = \boldsymbol{Z}, \\ & \boldsymbol{P}_t = \boldsymbol{R}_t, \quad t = 1, \ldots, T-1.\end{aligned}$$

The ADMM update steps to this program are a slight modification of (5.5)-(5.7). We distinguish f_0 from f_t and g_t since the former has separable variable structure, while the latter are functions that have a marginal coupling between the transport coupling $\boldsymbol{P}_t, \boldsymbol{R}_t$ and the marginal variables $\boldsymbol{w}_t, \boldsymbol{z}_t$, with slack variables $\boldsymbol{u}_t, \boldsymbol{v}_t$. f_0 is updated using a standard ADMM approach for solving RPCA (see [139, §5]). Given its separable nature, all f_t, g_t may be updated in parallel using the update step defined by (5.5) and (5.6) using unbalanced marginal constraints (with $p = 1$) of Section 5.1.4.

We demonstrate our framework's ability customize ground costs with the matrix

$$C_{ij} = \begin{cases} |\delta_i - \delta_j|^q, & \text{if } \ i, j \in \Omega \\ 10^4, & \text{otherwise} \end{cases},$$

with $q = 0.5$, and where δ_i denote support coordinates and Ω defines a user-defined sup-

port for which we provide two examples: (i) *passband support* $\Omega_{\text{pass}} := \{i \in \mathcal{Z} : f_{\text{low}} \leq i \leq f_{\text{high}}\}$, and (ii) *directional support* $\Omega_{\text{dir}} := \{i, j \in \mathcal{Z} : j \geq i\}$ as an example that biases up-chirps due to its triangular structure. In Figure 5.4, datasets `Wh_0066.wav` and `Wh_0082.wav` contain directional chirps so we apply both the passband and the directional filter (i.e., $\Omega = \Omega_{\text{pass}} \cap \Omega_{\text{dir}}$). Since dataset `Wh_0057.wav` contains bi-directional chirps, a directional filter does not apply so we only use a passband filter.

Figure 5.4 highlights our algorithm's superior qualitative performance over standard RPCA at separating the underlying chirp signal from background noise and interference. The OT-regularizer is shown to be a good fit for non-linear chirps of this nature since they model temporally-connected structure that conserves-mass over time. Additionally, with a carefully designed ground cost matrix, we may untangle chirps from vertical streaks of instantaneous energy, which a standard sparsity regularizer would be unable to do.

5.4 Discussion

In this chapter, we contributed a solver framework for the class of OT-regularized convex problems in Euclidean space. Our method jointly exploits simplex and Euclidean geometries to achieve efficient convergence using the Bregman ADMM framework [36]. We present an in depth discussion surrounding the similarities and differences between our approach and the Sinkhorn method proposed by [46], with versatility and accuracy being key advantages with our approach. Unlike interior-point methods, our method is parallelizable with efficient per-iteration cost, making it amenable for large-scale problems. In numerical tests, a GPU implementation of our method outperformed the highly-optimized commercial interior point solver Gurobi [62] with speedups of more than an order of magnitude. We showcased the flexibility of our framework with two types of OT-regularized convex problems. First, we consider OT-regularization for online dynamical estimation in a novel EEG source-localization application. Using synthetic trials, we demonstrated how an OT-regularizer can exploit temporal and physiological information to produce su-

perior estimations compared to standard approaches [169, 170]. Second, we consider a large-scale background-chirp separation problem for time-frequency data using a novel OT-regularized robust PCA method. Although this is a challenging problem that requires us to simultaneously solve a large number of OT problems (over 800 in our examples), our framework scales well because it is a parallelized distributed algorithm. We qualitatively illustrate how OT-regularization is successful at separating frequency-varying chirps from both low-rank interference and sparse instantaneous bursts of energy.

We propose several theoretical and applied lines of investigation for future work. On the theoretical front, we propose (i) performing analysis on the convergence rate [36] and contraction factors (e.g., using computer assisted proof methodologies [183]), and (ii) developing reconstruction error bounds on the proposed OT-regularization methods (e.g., balanced OT-regularized least-squares in Section 5.1.3). On the applied front, we propose (i) improving the convergence rate via acceleration techniques such as over-relaxation, or accelerated gradient methods [184], and (ii) developing multi-GPU architectures for large scale multi-OT distributed problems (e.g., the OT-regularized RPCA of Section 5.3.2).

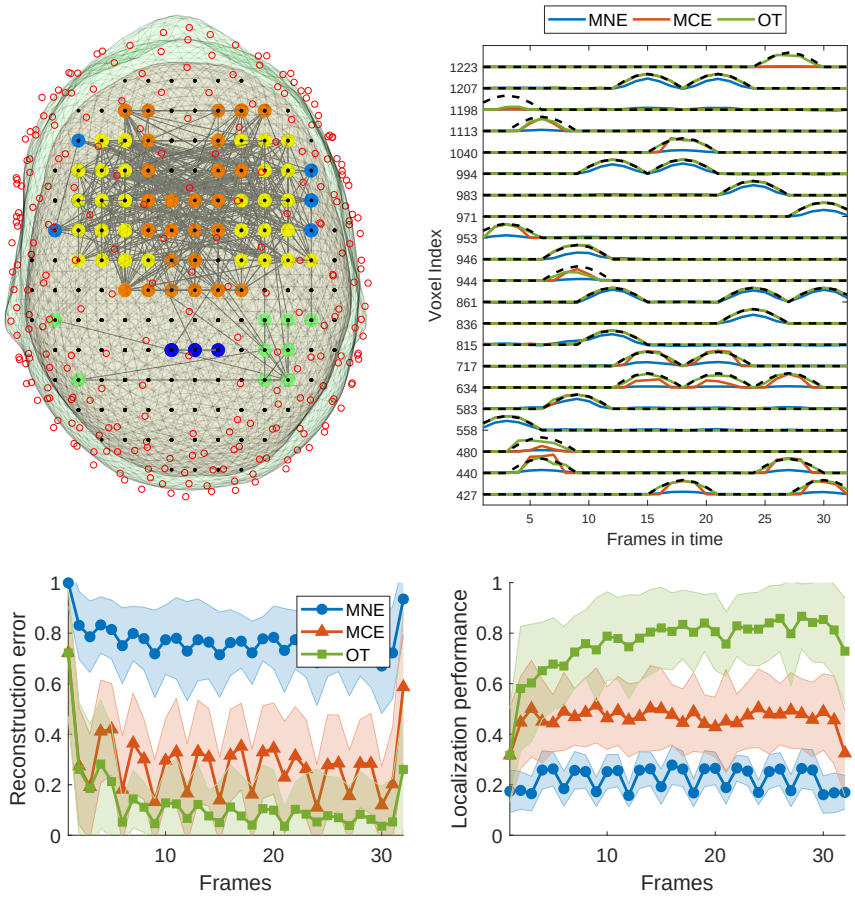


Figure 5.3: Reconstruction accuracy and localization performance in an EEG source localization application. *A priori* functional-network information may be incorporated via an optimal transport (OT) regularizer for dynamics-assisted inference. The top left image visualizes our EEG source localization problem, where red circles represent sensors on the scalp, black dots represent source dipole locations, colored markers represent the nodes within the default mode network, and grey-lines represent network connections. The top right plot visualizes time traces of the source activation of selected voxels within the network for a single trial. Ground truth activations are represented by black dotted lines. The bottom two plots aggregates results from 40 trials. OT-regularization improves qualitative and quantitative performance compared to non-dynamical counter-parts such as minimum norm estimation (MNE) [169] (i.e., Tikhonov-regularized least-squares) and the minimum current estimation (MCE) [170] (ℓ_1-regularized least squares).

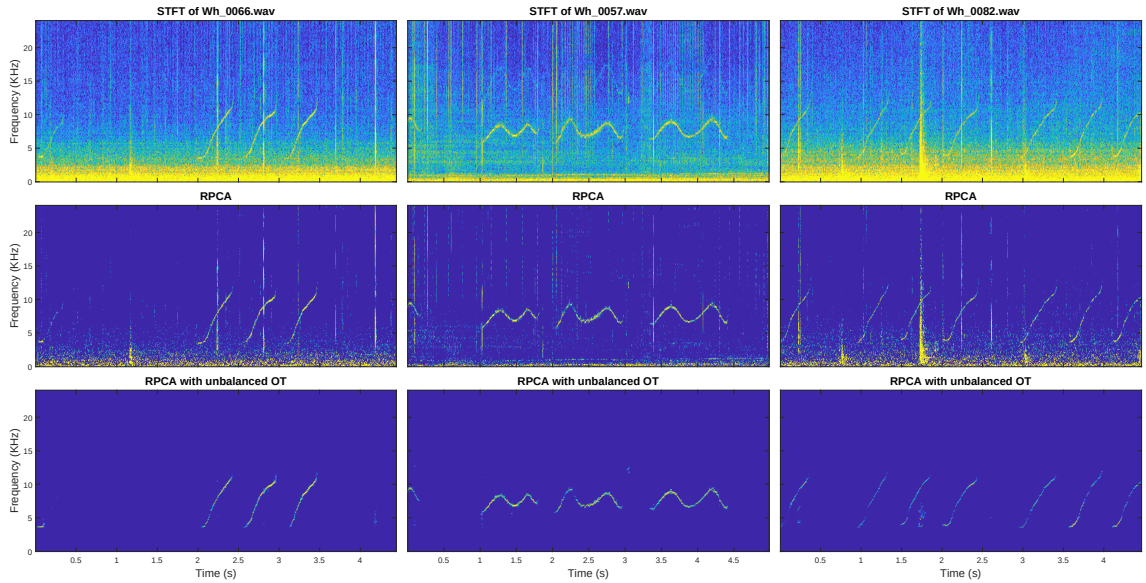


Figure 5.4: Separation of chirp signals from noise and interference in STFT magnitude data (plotted with log scaling). We present three examples of varying SINR (in respective columns). The first row shows the raw STFT of dolphin chirps contaminated with noise from the dataset in [178]. The second and third rows, respectively, show outputs from separation approaches: RPCA and our proposed algorithm. Our proposal of using an unbalanced optimal transport regularizer with RPCA demonstrates better qualitative separation performance than RPCA, which is especially effective at filtering away energy bursts (i.e., vertical 'streaks').

CHAPTER 6

HIERARCHICAL OPTIMAL TRANSPORT FOR DISTRIBUTION ALIGNMENT

In many machine learning applications, it is necessary to meaningfully aggregate, through alignment, different but related datasets (e.g., data across time points or under different conditions or contexts). Alignment is an important problem at the heart of transfer learning [185, 186], point set registration [187, 188, 189], and shape analysis [190, 191, 192], but is generally NP hard. In recent years, distribution alignment methods that use optimal transport (OT) have been shown to provide state-of-the-art transfer in domain adaptation tasks [51, 50]. Distribution alignment-based approaches cast alignment as an optimization problem that aims to match two distributions. However, when the source and target do not align exactly (e.g., noisy, undersampled) or have complicated multi-modal structure, algorithms suffer from poor local minima. Thus, leveraging additional structure in the problem is necessary to regularize OT and constrain the solution space.

Here, we leverage the fact that heterogeneous datasets often admit *clustered* or *multi-subspace* structure to improve distribution alignment. Our solution to this problem is to simultaneously estimate the cluster alignment across two datasets using their local geometry, while also solving a global alignment problem to meld these local estimates. We introduce a hierarchical formulation of OT for clustered and multi-subspace datasets called *Hierarchical Wasserstein Alignment (HiWA)*. We empirically show that when data can be well approximated with Gaussian mixture models (GMMs) or lie on a union of subspaces, we may leverage existing clustering pipelines (e.g., sparse subspace clustering [193]) to improve alignment. Our main contributions[1] are organized as follows:

[1]This work was performed in collaboration with Christopher J. Rozell, Max Dabagia, and Eva L. Dyer. MD was responsible for experiments on the neural decoding application. JL developed the algorithm, theoretically analyzed the algorithm, and tested the algorithm on synthetic data. ELD and CJR supervised the project.

Methodology development. We present the first distribution alignment method that exploits hierarchical structure. Our method relies on the Wasserstein distance to exploit geometry, relieving the need for kernel density estimation methods (e.g., Kullbeck-Liebler divergences or maximum mean discrepancy). We formulate a novel hierarchical Wasserstein objective function that extends the classical Wasserstein distance with hierarchical structure. As such, our method is effective at aligning multimodal data, which would otherwise entrap a non-hierarchical Wasserstein alignment method in local minima.

Distributed solver. To solve the problem numerically, we propose a distributed ADMM algorithm that exploits the Sinkhorn distance. By parallelizing computation across clusters, it has a computational complexity that scales quadratically with the size of the largest cluster (rather than with the size of the *entire* dataset as per vanilla Sinkhorn). In principle, this makes our solution tractable for datasets that contain many clusters.

Theoretical analyses. When the transformation between datasets is unitary, we provide analyses that reveal key geometric and sampling insights. Namely, we provide (i) uniqueness conditions on the clusters correspondences for two hierarchy levels, (ii) error bounds on our hierarchical Wasserstein objective, and (iii) the worst-case geometric dataset configurations for our alignment method.

Synthetic data and real data tests. To test and benchmark our approach, we applied it to synthetic data generated from mixtures of low-rank Gaussians and study the impact of different geometric properties of the data on alignment to confirm the predictions of our theoretical analysis. Next, we applied our approach to a neural decoding application where the goal is to predict movement directions from populations of neurons in the macaque primary motor cortex. Our results demonstrate that when clustered structure exists in neural datasets and is consistent across trials or time points, a hierarchical alignment strategy that leverages such structure can provide significant improvements in unsupervised decoding from ambiguous (symmetric) movement patterns.

6.1 Background and related work

6.1.1 Transfer learning and distribution alignment

A fundamental goal in transfer learning is to aggregate related datasets by learning an alignment between them. We wish to learn a transformation $T \in \mathcal{T}$, where $\mathcal{T}$ refers to some class of transformations that aligns distributions under a notion of probability divergence $\mathcal{D}(\cdot|\cdot)$ between a target distribution μ and a reference (source) distribution ν:

$$\min_{T \in \mathcal{T}} \mathcal{D}(T(\mu)|\nu). \tag{6.1}$$

Various probability divergences have been proposed in the literature, such as Euclidean least-squares (when data ordering is known) [194, 195, 196], Kullbeck-Liebler (KL) [197], maximum mean discrepancy (MMD) [198, 199, 200, 201], and the Wasserstein distance [51], where trade-offs are often statistical (e.g., consistency, sample complexity) versus computational. Alignment problems are ill-posed since the space of $\mathcal{T}$ is large, so *a priori* structure is often necessary to constrain $\mathcal{T}$ based on geometric assumptions. Compact manifolds like the Grassmann or Stiefel [202, 203] are primary choices when little information is present, as they preserve isometry. Non-isometric transformations, though richer, demand much more structure (e.g., manifold or graph structure) [204, 205, 52, 206, 51].

6.1.2 Low-rank and union of subspaces models

Principal components analysis (PCA), one of the most popular methods in data science, assumes a *low-rank* model where the top-k principal components of a dataset provide the optimal rank-k approximation under an Euclidean loss. This has been extended to robust (sparse errors) settings [193], and multi- (union of) subspaces settings where data can be partitioned into disjoint subsets where each subset of data is locally low-rank [207]. Transfer learning methods based on subspace alignment [208, 209, 210] work well with

zero-mean unimodal datasets, but struggle on more complicated modalities (e.g., Gaussian mixtures or union of subspaces) due to a mixing of covariances. Related to our work, [211] performs multi-subspace alignment by greedily assigning correspondences between subspaces using chordal distances; this however neglects sign ambiguities in principal directions since subspaces inadequately describe a distribution's shape.

6.1.3 Optimal transport

Optimal transport (OT) [212] is a natural type of divergence for registration problems because it accounts for the underlying geometry of the space. In Euclidean settings, OT is a metric known as the Wasserstein distance $\mathcal{W}(\mu, \nu)$ which measures the minimum effort required to "displace" points across measures μ and ν (understood here as empirical point clouds). Therefore, OT by design relieves the need for kernel estimation to create an overlapping support of the measures μ, ν. Despite this attractive property, it has both a poor numerical complexity of $O(n^3 \log n)$ (where n is the sample size) and a dimension-dependent sample complexity of $O(n^{-1/d})$, where the data dimension is d [66, 67]. Recently, an entropically regularized version of OT known as the Sinkhorn distance [63] has emerged as a compelling divergence measure; it not only inherits OT's geometric properties but also has superior computational and sample complexities of $O(n^2)$ and $O(n^{-1/2})$[2], respectively. It has also become a versatile building block in domain adaptation [51, 50]. Prior art [51] has largely exploited the OT's push-forward as the alignment map since this map minimizes the OT cost between the source and target distributions while allowing a priori structure to be easily incorporated (e.g., to preserve label/graphical integrity). Such an approach, however, is fundamentally expensive when $d \ll n$ since the primary optimization variable is a large transport coupling (i.e., $\mathbb{R}^{n \times n}$), while in reality the alignment mapping is merely $\mathbb{R}^d \mapsto \mathbb{R}^d$. Moreover, it assumes that the source and target distributions are close in terms of their squared Euclidean distance, but this does not generally hold in the alignment of

[2] Dependent on a regularization parameter [69].

arbitrary latent spaces.

6.2 Hierarchical Wasserstein alignment

6.2.1 Preliminaries and notation

Consider clustered datasets $\{\boldsymbol{X}_i \in \mathbb{R}^{D\times n_{x,i}}\}_{i=1}^S$ and $\{\boldsymbol{Y}_j \in \mathbb{R}^{D\times n_{y,j}}\}_{j=1}^S$ whose clusters are denoted with the indices i, j and whose columns are treated as $\mathbb{R}^D$ embedding coordinates. Let $n_{x,i}$ ($n_{y,j}$) denote the number of samples in the i-th (j-th) cluster of dataset $\boldsymbol{X}$ (dataset $\boldsymbol{Y}$). We respectively express the empirical measures of clusters $\boldsymbol{X}_i$ and $\boldsymbol{Y}_j$ as $\mu_i := \frac{1}{n_{x,i}} \sum_{k=1}^{n_{x,i}} \delta_{\boldsymbol{X}_i(k)}$ and $\nu_j := \frac{1}{n_{y,j}} \sum_{l=1}^{n_{y,j}} \delta_{\boldsymbol{Y}_j(l)}$, where $\delta_{\boldsymbol{x}}$ refers to a point mass located at coordinate $\boldsymbol{x} \in \mathbb{R}^D$. The squared 2-Wasserstein distance between μ_i and ν_j is defined as

$$\mathcal{W}_2^2(\mu_i, \nu_j) := \min_{\boldsymbol{Q}\in\mathcal{U}(n_{x,i},n_{y,j})} \sum_{k=1}^{n_{x,i}} \sum_{l=1}^{n_{y,j}} Q(k,l) \left\|\boldsymbol{X}_i(k) - \boldsymbol{Y}_j(l)\right\|_2^2$$

where $\boldsymbol{Q}$ is a doubly stochastic matrix that encodes point-wise correspondences (i.e., the (k,l)-th entry describes the flow of mass between $\delta_{\boldsymbol{X}_i(k)}$ and $\delta_{\boldsymbol{Y}_j(l)}$), $\boldsymbol{X}_i(k)$ is the k-th column of matrix $\boldsymbol{X}_i$, and the constraint $\mathcal{U}(m,n) := \{\boldsymbol{Q} \in \mathbb{R}_+^{m\times n} : \boldsymbol{Q}\mathbb{1}_n = \mathbb{1}_m/m, \boldsymbol{Q}^\top \mathbb{1}_m = \mathbb{1}_n/n\}$ refers to the *uniform* transport polytope (with $\mathbb{1}_m$ a length m vector containing ones).

6.2.2 Overview

Although unsupervised alignment is challenging due to the presence of local minima, the imposition of additional structure will help to prune them away. Our key insight is that hierarchical structure decomposes a complicated optimization surface into simpler ones that are less prone to local minima. We formulate a hierarchical Wasserstein approach to align datasets with known (or estimated) clusters $\{\mu_i\}_{i=1}^S, \{\nu_j\}_{j=1}^S$ but whose correspondences are unknown. The task therefore is to jointly learn the alignment T and the cluster-

correspondences:

$$\min_{\boldsymbol{P}\in\mathcal{B}_S, T\in\mathcal{T}} \sum_{i=1}^{S}\sum_{j=1}^{S} P_{ij}\mathcal{W}_2^2(T(\mu_i),\nu_j), \tag{6.2}$$

where the matrix $\boldsymbol{P}$ encodes the strength of correspondences between clusters, with a large P_{ij} value indicating a correspondence between clusters i, j, and a small value indicating a lack thereof. We note that $\mathcal{B}_S := \mathcal{U}(S, S)$ is a special type of transport polytope known as the S-th Birkhoff polytope. Interestingly, this becomes a nested (or block) OT formulation, where correspondences are resolved at two levels: the outer level resolves cluster-correspondences (via $\boldsymbol{P}$) while the inner level resolves point-wise correspondences between cluster points (via the Wasserstein distance). Refer to Figure 6.1 for a high level illustration of the method on a simplified example.

Alignment over the Stiefel manifold. Assuming clusters lie on subspaces and principal angles between subspaces are "well preserved" across $\boldsymbol{X}$ and $\boldsymbol{Y}$ (we make this precise in Theorem 6.3.2), an isometric transformation suffices. Hence, we solve (6.2) with $\mathcal{T} \leftarrow \mathcal{V}_{D,D}$, which refers to the Stiefel manifold defined as $\mathcal{V}_{k,d} := \{\boldsymbol{R} \in \mathbb{R}^{k\times d} : \boldsymbol{R}^\top\boldsymbol{R} = \boldsymbol{I}_d\}$ and $\boldsymbol{I}_d$ refers to the $d \times d$ identity matrix. Explicitly, we have

$$\min_{\boldsymbol{P},\boldsymbol{R},\{\boldsymbol{Q}_{ij}\}} \sum_{i,j} P_{ij}C_{ij}(\boldsymbol{R},\boldsymbol{Q}_{ij}) \quad \text{s.t.} \quad \boldsymbol{P}\in\mathcal{B}_S, \quad \boldsymbol{R}\in\mathcal{V}_{D,D}, \quad \boldsymbol{Q}_{ij}\in\mathcal{U}(n_{x,i}, n_{y,j}). \tag{6.3}$$

Here, C_{ij} measures pairwise cluster divergences using the squared 2-Wasserstein distance under a Stiefel transformation $\boldsymbol{R}$ acting on cluster i, i.e.,

$$C_{ij}(\boldsymbol{R},\boldsymbol{Q}_{ij}) := \frac{1}{D}\sum_{k,l} \boldsymbol{Q}_{ij}(k,l)\,\|\boldsymbol{R}\boldsymbol{X}_i(k) - \boldsymbol{Y}_j(l)\|_2^2\,. \tag{6.4}$$

Finally, we include entropic regularization over transportation couplings $\boldsymbol{P}$ and all $\boldsymbol{Q}_{ij}$'s to modify the Wasserstein distances to Sinkhorn distances, so as to take advantage of its superior computational and sample complexities. Omitting constraints for brevity, our

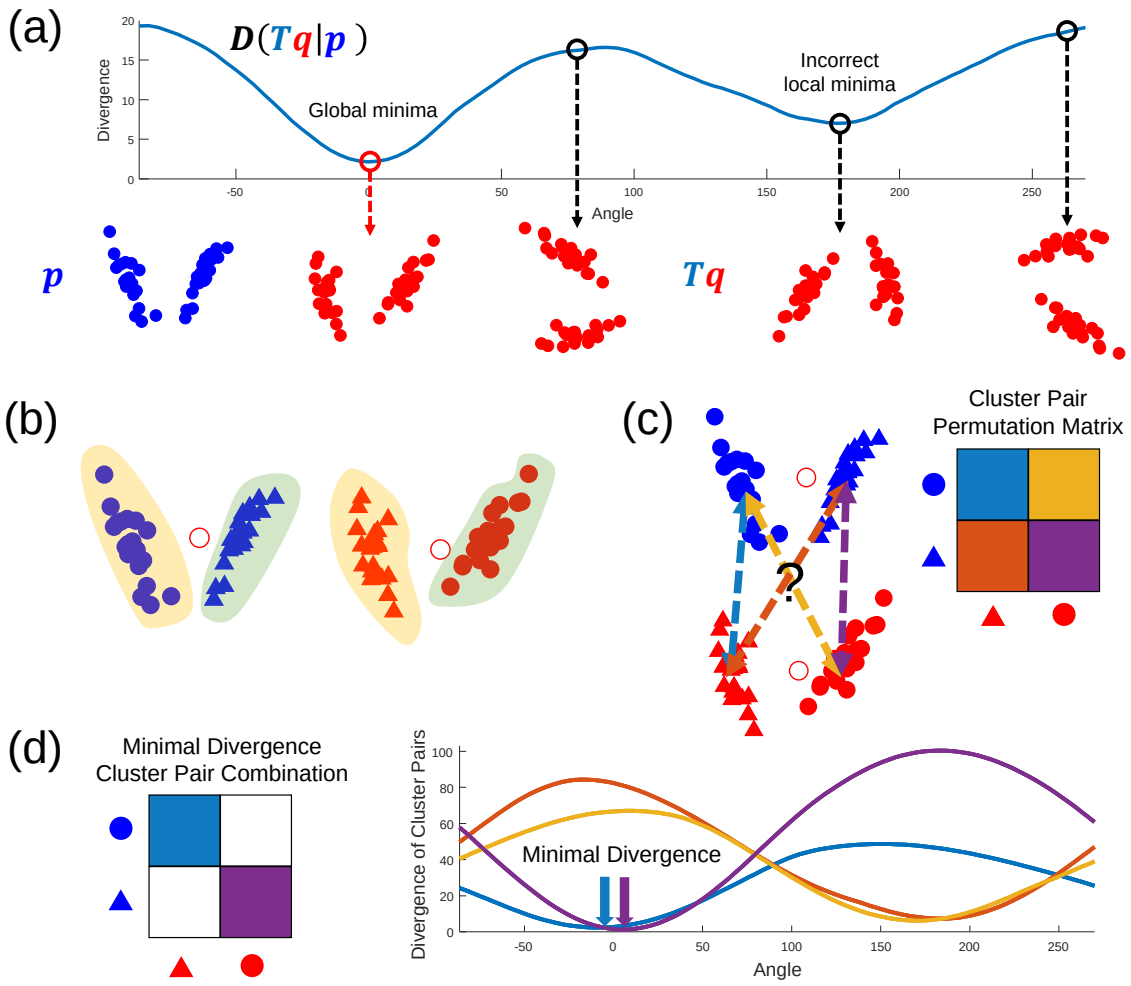


Figure 6.1: Overview of Hierarchical Wasserstein Alignment. (a) The Wasserstein distance is a divergence measure between point clouds Tq and p. The goal is to find the rotation T that produces the minimum divergence between Tq and p. The search is challenging due to presence of multiple local minima. (b) Our approach assumes that individual datasets can be decomposed into their individual clusters, e.g., using unsupervised approaches such as sparse subspace clustering. (c) The goal becomes the search for pairs of clusters that match (i.e., blue circle to red triangle, blue triangle to red circle). (d) This combinatorial search can be framed as finding the two pairs that yield overall minimal summed divergence. The pairwise divergence landscape exhibits convex-like properties, making it an easier problem than the original one.

final problem is given as

$$\min_{\boldsymbol{P},\boldsymbol{R},\{\boldsymbol{Q}_{ij}\}} \sum_{i,j} \Big(P_{ij} C_{ij}(\boldsymbol{R}, \boldsymbol{Q}_{ij}) + H_{\gamma_2}(\boldsymbol{Q}_{ij}) \Big) + H_{\gamma_1}(\boldsymbol{P}), \tag{6.5}$$

where $\gamma_1, \gamma_2 > 0$ are the entropic regularization parameters and the negative entropy function is defined as $H_\gamma(\boldsymbol{P}) := \gamma \sum_{i,j} P_{ij} \log P_{ij}$. Parameters γ_1, γ_2 control the correspondence entropy, therefore (6.5) approximates (6.3) when $\gamma_1, \gamma_2 > 0$, but reverts to the original problem (6.3) as $\gamma_1, \gamma_2 \to 0$.

6.2.3 Distributed non-convex ADMM algorithm

Problem (6.5) is non-convex due to multilinearity in the objective and its Stiefel manifold domain. It has recently been shown that the augmented directions method of multipliers (ADMM) [28, 22] can be globally convergent even in non-convex settings [32]. Furthermore, since (6.5) readily admits a splitting structure that separates the individual C_{ij} blocks, we develop a distributed ADMM solver. We proceed to split (6.5) as follows:

$$\min_{\boldsymbol{P},\boldsymbol{R},\{\boldsymbol{R}_{ij},\boldsymbol{Q}_{ij}\}} \sum_{i,j} \Big(P_{ij} C_{ij}(\boldsymbol{R}_{ij}, \boldsymbol{Q}_{ij}) + H_{\gamma_2}(\boldsymbol{Q}_{ij}) \Big) + H_{\gamma_1}(\boldsymbol{P}) \quad \text{s.t.} \quad \boldsymbol{R} = \boldsymbol{R}_{ij}, \;\; \forall i, j,$$

noting that the set constraints are omitted for brevity. Its augmented Lagrangian is

$$\mathcal{L}_\mu = \sum_{i,j} \Big(P_{ij} C_{ij}(\boldsymbol{R}_{ij}, \boldsymbol{Q}_{ij}) + \frac{\mu}{2D} \|\boldsymbol{R}_{ij} - \boldsymbol{R} + \boldsymbol{\Lambda}_{ij}\|_F^2 + H_{\gamma_2}(\boldsymbol{Q}_{ij}) \Big) + H_{\gamma_1}(\boldsymbol{P}),$$

where $\mu > 0$ is the ADMM parameter and $\{\Lambda_{ij}\}$ are Lagrange multipliers.

ADMM admits the following sequence of updates:

$$(\boldsymbol{R}_{ij}^{(t+1)}, \boldsymbol{Q}_{ij}^{(t+1)}) \leftarrow \underset{\substack{\boldsymbol{R}_{ij} \in S(D,D) \\ \boldsymbol{Q}_{ij} \in \mathcal{U}(n_{x,i}, n_{y_j})}}{\arg\min} \quad P_{ij}^{(t)} C_{ij}(\boldsymbol{R}_{ij}, \boldsymbol{Q}_{ij}) + \frac{\mu}{2D} \|\boldsymbol{R}_{ij} - \boldsymbol{R}^{(t)} + \boldsymbol{\Lambda}_{ij}^{(t)}\|_F^2 + H_{\gamma_2}(\boldsymbol{Q}_{ij}), \tag{6.6}$$

$$\boldsymbol{P}^{(t+1)} \leftarrow \underset{\boldsymbol{P} \in \mathcal{B}_c}{\arg\min} \quad \sum_{i,j} P_{ij} C_{ij}(\boldsymbol{R}_{ij}^{(t+1)}, \boldsymbol{Q}_{ij}^{(t+1)}) + H_{\gamma_1}(\boldsymbol{P}), \tag{6.7}$$

$$\boldsymbol{R}^{(t+1)} \leftarrow \underset{\boldsymbol{R} \in S(D,D)}{\arg\min} \quad \sum_{i,j} \|\boldsymbol{R}_{ij}^{(t+1)} - \boldsymbol{R} + \boldsymbol{\Lambda}_{ij}^{(t)}\|_F^2, \tag{6.8}$$

$$\boldsymbol{\Lambda}_{ij}^{(t+1)} \leftarrow \boldsymbol{\Lambda}_{ij}^{(t)} + \boldsymbol{R}_{ij}^{(t+1)} - \boldsymbol{R}. \tag{6.9}$$

Update (6.6) involves an alternating minimization over $\boldsymbol{Q}_{ij}$ and $\boldsymbol{R}_{ij}$ whereby the first variable is fixed while the second is minimized, followed by the second fixed and the first minimized, and the procedure is repeated until convergence is achieved. When solving for $\boldsymbol{R}_{ij}$ we have the following Stiefel manifold optimization:

$$\begin{aligned} \boldsymbol{R}_{ij}^{(t+1)} &\leftarrow \underset{\boldsymbol{R}_{ij} \in S(D,D)}{\arg\min} \quad P_{ij}^{(t)} C_{ij}(\boldsymbol{R}_{ij}, \boldsymbol{Q}_{ij}) + \frac{\mu}{2D} \|\boldsymbol{R}_{ij} - \boldsymbol{R}^{(t)} + \boldsymbol{\Lambda}_{ij}^{(t)}\|_F^2 \\ &= \underset{\boldsymbol{R}_{ij} \in S(D,D)}{\arg\max} \quad \operatorname{tr}\left(\left(2P_{ij}^{(t)} \boldsymbol{Y}_j \boldsymbol{Q}_{ij}^\top \boldsymbol{X}_i^\top + \mu(\boldsymbol{R}^{(t)} - \boldsymbol{\Lambda}_{ij}^{(t)}) \right) \boldsymbol{R}_{ij}^\top \right) = \boldsymbol{U}\boldsymbol{V}^\top, \end{aligned}$$

where $2P_{ij}^{(t)} \boldsymbol{Y}_j \boldsymbol{Q}_{ij}^\top \boldsymbol{X}_i^\top + \mu(\boldsymbol{R}^{(t)} - \boldsymbol{\Lambda}_{ij}^{(t)}) = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$ is its SVD. We employ the Sinkhorn algorithm (Algorithm 1 of [63]) to solve for $\boldsymbol{Q}_{ij}$, using an entropic parameter of $\gamma_2 / P_{ij}^{(t)}$ and uniform marginals. $\boldsymbol{R}_{ij}^{(t+1)}$ and $\boldsymbol{Q}_{ij}^{(t+1)}$ are retrieved once the alternating minimization converges.

Update (6.7) also employs the Sinkhorn algorithm over the cost matrix generated by $C_{ij}(\boldsymbol{R}_{ij}^{(t+1)}, \boldsymbol{Q}_{ij}^{(t+1)})$ using variables found in update (6.6), along with an entropic parameter of γ_1 and uniform marginals.

Update (6.8) is a consensus update over a Stiefel manifold optimization:

$$\boldsymbol{R}^{(t+1)} \leftarrow \underset{\boldsymbol{R}\in S(D,D)}{\arg\min} \sum_{i,j} \|\boldsymbol{R}_{ij}^{(t+1)} - \boldsymbol{R} + \boldsymbol{\Lambda}_{ij}^{(t)}\|_F^2 = \boldsymbol{U}\boldsymbol{V}^\top,$$

where $\sum_{i,j} \boldsymbol{R}_{ij}^{(t+1)} + \boldsymbol{\Lambda}_{ij}^{(t)} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$ is its SVD.

The algorithm may be summarized in two steps: (i) a distributed step that asks all cluster pairs to individually find their optimal transformations $\boldsymbol{R}_{ij}$ in parallel, and (ii) a consensus step that aggregates all the found transformations according to a weighting that is proportional to correspondence strengths P_{ij}. Algorithm 5 summarizes our method.

Algorithm 5 Hierarchical Wasserstein Alignment (HiWA) Algorithm

procedure HIERARCHICALWASSERSTEINALIGNMENT($\gamma_1, \gamma_2, \mu, \{\boldsymbol{X}_i\}_{i=1}^S, \{\boldsymbol{Y}_j\}_{j=1}^S$)
  $\boldsymbol{R} \leftarrow$ random $\mathcal{V}_{D,D}$, $\boldsymbol{P} \leftarrow \mathbb{1}_S\mathbb{1}_S^\top/S^2$, $\boldsymbol{\Lambda}_{ij} \leftarrow \boldsymbol{0}$, $\forall i,j$ ▷ Initialization
  while not converged **do**
    for all i, j in parallel **do**
      $\boldsymbol{Q}_{ij} \leftarrow \mathbb{1}_{n_{x,i}}\mathbb{1}_{n_{y,j}}^\top/n_{x,i}n_{y,j}$
      while not converged **do**
        $\boldsymbol{R}_{ij} \leftarrow$ STIEFELALIGNMENT$(2P_{ij}\boldsymbol{Y}_j\boldsymbol{Q}_{ij}^\top\boldsymbol{X}_i^\top + \mu(\boldsymbol{R} - \boldsymbol{\Lambda}_{ij}))$
        $\boldsymbol{Q}_{ij} \leftarrow$ SINKHORN$(\gamma_2/P_{ij}, \boldsymbol{C}(k,l) \leftarrow \frac{1}{D}\|\boldsymbol{R}_{ij}\boldsymbol{X}_i(k) - \boldsymbol{Y}_j(l)\|_2^2)$
      end while
    end for
    $\boldsymbol{P} \leftarrow$ SINKHORN$(\gamma_1, \boldsymbol{C}(i,j) \leftarrow C_{ij}(\boldsymbol{R}_{ij}, \boldsymbol{Q}_{ij}))$
    $\boldsymbol{R} \leftarrow$ STIEFELALIGNMENT$(\sum_{i,j} \boldsymbol{R}_{ij} + \boldsymbol{\Lambda}_{ij})$
    $\boldsymbol{\Lambda}_{ij} \leftarrow \boldsymbol{\Lambda}_{ij} + \boldsymbol{R}_{ij} - \boldsymbol{R}$, $\forall i,j$
  end while
end procedure

procedure SINKHORN($\gamma, \boldsymbol{C} \in \mathbb{R}^{m\times n}$)
  $\boldsymbol{K} \leftarrow \exp(-\boldsymbol{C}/\gamma)$, $\boldsymbol{v} \leftarrow \frac{\mathbb{1}_n}{n}$
  while not converged **do**
    $\boldsymbol{u} \leftarrow \frac{\mathbb{1}_m}{m} \oslash \boldsymbol{K}\boldsymbol{v}$
    $\boldsymbol{v} \leftarrow \frac{\mathbb{1}_n}{n} \oslash \boldsymbol{K}^\top\boldsymbol{u}$
  end while
  $\boldsymbol{P} \leftarrow \mathrm{diag}(\boldsymbol{u})\boldsymbol{K}\,\mathrm{diag}(\boldsymbol{v})$
end procedure

procedure STIEFELALIGNMENT($\boldsymbol{A}$)
  $(\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}) \leftarrow$ SVD$(\boldsymbol{A})$
  $\boldsymbol{R} \leftarrow \boldsymbol{U}\boldsymbol{V}^\top$
end procedure

Notation:
$\oslash$: elementwise division
$\exp(\cdot)$: elementwise exponential
$\mathrm{diag}(\cdot)$: diagonal matrix of argument

Parameters. Entropic parameters γ_1, γ_2 relax the one-to-one cluster correspondence assumption, balancing a trade off between alignment precision (small γ) and sample com-

plexity (large γ). Numerically, negative entropy adds strong convexity to the program, reducing sensitivity towards perturbations at the cost of a slower convergence rate. The ADMM parameter μ controls the 'strength' of the consensus, or from an algorithmic viewpoint, the gradient step size.

Distributed consensus. Update steps (6.6) and (6.9) can be performed in parallel over all cluster pairs (i.e., S^2 in total), making it amenable for parallel implementation. When fully parallelized, the algorithm has a per-iteration computational complexity of $O(n_i n_j)$, where n_i, n_j refers to the number of points in the largest clusters of $\boldsymbol{X}, \boldsymbol{Y}$ respectively (compared to vanilla Sinkhorn's $O(n_x n_y)$ complexity where n_x, n_y refers to the *total* number of points in respective datasets, assuming $D \ll \max(n_i, n_j)$).

Stopping criteria. In lines 3 and 6 of Algorithm 5, possible stopping criteria are (i) $\|\boldsymbol{R}^{(t+1)} - \boldsymbol{R}^{(t)}\|_F \leq \tau$ where the difference is between the current and previous iteration's transformation and τ is the tolerance, and (ii) $t \leq T$ where T is the maximum number of iterations.

Robustness against initial conditions. We build in robustness against initial conditions by ordering update (6.6) before (6.7) such that when μ is sufficiently small, the ADMM sequence is influenced more by the data (i.e., first term of (6.6)) than by initial conditions (i.e., second term).

6.3 Theoretical guarantees

While the previous section explains *how* to align clustered datasets, in this section, we aim to answer the question of *when* and *how well* they can be aligned. We provide necessary conditions for *cluster-based alignability* as well as *alignment perturbation bounds* according to problem (6.3)'s formulation. To simplify our analysis, we make the following assumptions: (i) each of the clusters contain the same number of datapoints n, (ii) the ground truth cluster correspondences are $\boldsymbol{P}^\star = \boldsymbol{I}_S/S$ (i.e., diagonal containing $1/S$). Detailed proofs are given in the annexes.

6.3.1 Criterion for existence of unique cluster correspondences

he following result is a criterion that, if met, ensures the existence of the cluster-correspondence global minima $\boldsymbol{P}^\star$. This criterion requires that matched clusters must be closer in Wasserstein distance than mismatched clusters, according to a threshold according to Wasserstein's sample complexity (i.e., an asymptotic rate dependent on the clusters' *sample sizes* and *intrinsic dimensions*). Since these sample complexity results are based on the Wasserstein distance, we expect a less stringent criterion when using the Sinkhorn distance in (6.5) (due to superior sample complexity [69]).

Theorem 6.3.1 (Correspondence disambiguity criterion)**.** *Let all clusters be strictly low-rank where the dimension of the i-th cluster in the x-th dataset is $d_{x,i}$. Let $d_{x,i}, d_{y,j} > 4, \forall i, j \in [\![S]\!]$. Define $\widehat{C}^\star_{ij} := \min_{\boldsymbol{R}\in\mathcal{V}_{D,D},\boldsymbol{Q}_{ij}\in\mathcal{B}_n} C_{ij}(\boldsymbol{R}, \boldsymbol{Q}_{ij})$. Problem* (6.3) *yields the solution $\boldsymbol{P}^\star = \boldsymbol{I}_S/S$ with probability at least $1-\delta$ if, $\forall i, j : i \neq j$, the following criterion is satisfied:*

$$\widehat{C}^\star_{ij} + \widehat{C}^\star_{ji} - \widehat{C}^\star_{ii} - \widehat{C}^\star_{jj} > B_{x,i}(\delta) + B_{y,i}(\delta) + B_{x,j}(\delta) + B_{y,j}(\delta)$$

$$\textit{where } B_{z,k}(\delta) := c_{z,k} n^{-\frac{2}{d_{z,k}}} + \sqrt{\log(1/\delta)/2n}, \qquad c_{z,k} = 1458\Big(2 + \frac{1}{3^{d_{z,k}/2-2}-1}\Big).$$

Proof sketch. The proof contains two parts. In the first part, we consider perturbation conditions of the cost matrix $\boldsymbol{C}$ in a (non-variational) optimal transport program over the Birkhoff polytope. To be unperturbed from $\boldsymbol{P}^\star = \boldsymbol{I}_S/S$, we require that $C_{ij} + C_{ji} - C_{ii} - C_{jj} > 0, \forall i, j : i \neq j$. In the second part, we extend this condition to the the finite-sample regime by utilizing recently developed concentration bounds [67] for the p-Wasserstein distance, which essentially raises the disambiguity lower bound due to finite-sample uncertainty. □

6.3.2 Error bound on recovery of transformation under Stiefel constraints

Now, even if we have the global correspondence solution $\boldsymbol{P}^\star$, we still do not have the full picture about the alignment's quality. For example, all matching clusters may have very similar covariances, but principal angles between the clusters are "distorted" across the datasets. Our next theorem gives us an upper bound on the alignment error (for unitary transformations), and makes precise the notion of *global structure distortion*.

Theorem 6.3.2 (Cluster-based alignment perturbation bounds)**.** *Consider data matrices* $\{\boldsymbol{X}_i, \boldsymbol{Y}_i \in \mathbb{R}^{D\times n}\}_{i=1}^c$ *with known point-wise correspondence matrices* $\{\boldsymbol{Q}_{ii} \in \mathcal{B}_n\}_{i=1}^c$. *Define matrices*

$$\boldsymbol{X} := [\boldsymbol{X}_1\boldsymbol{Q}_{11}, \boldsymbol{X}_2\boldsymbol{Q}_{22}, \ldots, \boldsymbol{X}_c\boldsymbol{Q}_{cc}], \qquad \boldsymbol{Y} := [\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_c].$$

Set $\varepsilon^2 := \left\|\boldsymbol{Y}^\top\boldsymbol{Y} - \boldsymbol{X}^\top\boldsymbol{X}\right\|_F$. *If the criterion stated in theorem 6.3.1 is satisfied,* $\boldsymbol{X}$ *is full row rank, and* $\|\boldsymbol{X}^\dagger\|\varepsilon \leq \frac{1}{\sqrt{2}}(\|\boldsymbol{X}\|\,\|\boldsymbol{X}^\dagger\|)^{-1/2}$ *where* $\|\cdot\|$ *is the operator norm and* $\boldsymbol{X}^\dagger$ *is the pseudo-inverse of* $\boldsymbol{X}$*, then*

$$\min_{\boldsymbol{P}\in\mathcal{B}_c,\boldsymbol{R}\in\mathcal{V}_{D,D}} \sum_{i,j} P_{ij}C_{ij}(\boldsymbol{R}) \quad \leq \quad (\|\boldsymbol{X}\|\,\|\boldsymbol{X}^\dagger\| + 2)^2\|\boldsymbol{X}^\dagger\|^2\varepsilon^4 + D,$$

where $D = \sum_{i=1}^c \operatorname{tr}(\boldsymbol{X}_i(\boldsymbol{I}/n - \boldsymbol{Q}_{ii}\boldsymbol{Q}_{ii}^\top)\boldsymbol{X}_i^\top + (1/n-1)\boldsymbol{Y}_i\boldsymbol{Y}_i^\top)$ *is a data-dependent constant.*

Proof sketch. We utilize a recent perturbation result on the Procrustes problem (on a Frobenius norm objective) by Arias-Castro et al. [213] and adapt it to our squared 2-Wasserstein objective. □

We point out that ε plays a major role in the alignment error bound and quantifies the notion of *global structure distortion*. It therefore allows us to understand on how phenomena like covariate shift or misclustering impacts alignment. To shed some light in this regard,

we consider a simple analysis on a cluster-pair's error contribution to ε. Consider the decomposition of the (i,j)-th block of the Grammians related to clusters i and j, where their respective singular value decompositions are $\boldsymbol{X}_i\boldsymbol{Q}_{ii} = \boldsymbol{A}_i\boldsymbol{\Sigma}_{x,i}\boldsymbol{V}^\top$ and $\boldsymbol{Y}_j = \boldsymbol{B}_j\boldsymbol{\Sigma}_{y,j}\boldsymbol{V}^\top$. Defining the *blockwise* error between clusters i,j as

$$\varepsilon_{ij} := \left\|\boldsymbol{Y}_i^\top\boldsymbol{Y}_j - \boldsymbol{Q}_{ij}^\top\boldsymbol{X}_i^\top\boldsymbol{X}_j\boldsymbol{Q}_{jj}\right\|_F = \left\|\boldsymbol{\Sigma}_{y,i}\boldsymbol{B}_i^\top\boldsymbol{B}_j\boldsymbol{\Sigma}_{y,j} - \boldsymbol{\Sigma}_{x,i}\boldsymbol{A}_i^\top\boldsymbol{A}_j\boldsymbol{\Sigma}_{x,j}\right\|_F,$$

two components stand out: (i) *angular shift*, which is characterized by differences in principal angles between $\boldsymbol{B}_i^\top\boldsymbol{B}_j$ and $\boldsymbol{A}_i^\top\boldsymbol{A}_j$, and (ii) *spectral shift*, which is characterized by differences in spectra.

6.3.3 Worst-case geometric dataset conditions

Finally, we show that the subspace configuration of a dataset's clusters can also affect alignment. Pretend for a moment that external alignment information were present to aid in the disambiguation between two clusters. The following lemma tells us when such information is useless.

Lemma 6.3.3 (Uninformative alignment)**.** *Consider clusters $\boldsymbol{X}_i, \boldsymbol{Y}_j \in \mathbb{R}^{D\times n}$ and known point-wise correspondences $\boldsymbol{Q}_{ij} \in \mathcal{U}(n,n)$. Denote the left and right singular vectors of $\boldsymbol{Y}_j\boldsymbol{Q}_{ij}^\top\boldsymbol{X}_i^\top$ associated with the non-zero singular values as $\widetilde{\boldsymbol{U}}, \widetilde{\boldsymbol{V}} \in \mathbb{R}^{D\times r}$ with $r \leq D$. Define the set of orthogonal transformations that are constrained to agree with known angular directions as*

$$\mathcal{T}(\boldsymbol{U}', \boldsymbol{V}') := \{\boldsymbol{R} \in \mathbb{R}_+^{D\times D} : \boldsymbol{R}^\top\boldsymbol{R} = \boldsymbol{I}, \boldsymbol{R}\boldsymbol{V}' = \boldsymbol{U}'\},$$

where $\boldsymbol{U}', \boldsymbol{V}' \in \mathcal{V}_{D,r}$ with $r \leq D$. Given $\boldsymbol{U}', \boldsymbol{V}' \in \mathbb{R}^{D\times r'}$ with $r' \leq D$, we have

$$\min_{\boldsymbol{R}\in\mathcal{T}(\boldsymbol{U}',\boldsymbol{V}')} C_{ij}(\boldsymbol{R}) \geq \min_{\boldsymbol{R}\in\mathcal{V}_{D,D}} C_{ij}(\boldsymbol{R}), \tag{6.10}$$

with equality holding when $\langle \widetilde{\boldsymbol{U}}, \boldsymbol{U}' \rangle = \langle \widetilde{\boldsymbol{V}}, \boldsymbol{V}' \rangle$.

Direct consequences of this lemma are the following: When a dataset has equally-spaced subspaces, it has a maximally uninformative geometric configuration since angular information from other clusters (i.e., $\boldsymbol{U}', \boldsymbol{V}'$) can never increase the inter-cluster distance C_{ij} (i.e., equality in (6.10) always holds); it is hence a worst-case scenario for alignment. This also explains why alignment in very high-dimensional space is harder: All subspaces may be orthogonal to each other, and hence offer no "geometric" advantage in the joint alignment effort.

6.4 Numerical experiments on low-rank Gaussian mixtures

In this section, we validate our method as well as demonstrate its limiting characteristics under symmetric-subspace and finite-sample regimes. To generate our synthetic data, we repeat the following procedure for S clusters. We first randomly generate Gaussian distribution parameters $\mu_i \in \mathbb{R}^d, \Sigma_i \in \mathbb{R}^d : \Sigma_i \succeq 0$ (positive semi-definite), then randomly sample n data-points from these parameters, and finally project them into a random subspace $\boldsymbol{V}_i \in \mathbb{R}^{D \times d}$ in a $D > d$ dimensional embedding. We assume that the respective clusters are known, but the cluster-correspondences between datasets is not. We measure performance with two metrics: (i) *alignment error*, defined as the relative difference between the recovered versus true rotation acting on the data $\|\widehat{\boldsymbol{R}}\boldsymbol{X} - \boldsymbol{R}^\star\boldsymbol{X}\|_F^2 / \|\boldsymbol{R}^\star\boldsymbol{X}\|_F^2$, and (ii) *correpondence error*, defined as the sum of absolute differences between the recovered and the true correspondences $\sum_{ij} |\widehat{\boldsymbol{P}} - \boldsymbol{P}^\star|_{ij}$.

In Figure 6.2(a-b), we empirically validate the fact that equally spaced subspaces are indeed the worst-case scenario in alignment, as exposed by Lemma 6.3.3. We run our proposed algorithm on two identical datasets generated with parameters $S = 5, d = 2, D = 6$; the key difference being that one dataset has equally-spaced subspaces with a subspace similarity of $\|\boldsymbol{V}_i^\top \boldsymbol{V}_j\| = 1, \forall i \neq j$, while the other contains subspaces that are randomly selected on the Grassman manifold. We observe that equally-spaced subspaces have sig-

nificantly inferior performance compared to randomly-spaced subspaces, across various n. Interestingly, correspondence error is more tolerant than alignment towards subspace spacing configuration.

In Figure 6.2(c-d), we empirically study the effect of dimensions d and sample size n on the accuracy of alignment. We run our proposed algorithm on various dataset conditions by varying parameters $d = \{2, 3, 4, 5\}, n = \{12, 25, 50, 100, 200\}$ while approximately maintaining the average subspace correlations (i.e., $\mathbb{E}\|\boldsymbol{V}_i^\top \boldsymbol{V}_j\|_2$) by tuning D to control for subspace spacing biases, and fixing the cluster size $S = 5$. Both errors demonstrate sample complexities that are better than the theoretical $O(n^{-1/d})$, with correspondence error exhibiting greater robustness. We hypothesize this is due to the Sinkhorn distance's superior sample complexity.

In Figure 6.2(e-f), we evaluate our algorithm against benchmark methods in transfer learning and point set registration under two settings (50 trials, no random restarts permitted): a simple one in low-d (e) and a harder one in higher-d (f). Specifically, we compare HiWA when clusters are known (but correspondences not), HiWA with clustering via sparse subspace clustering [193] (HiWA-SSC), a Wasserstein alignment variant with no cluster-structure (WA), subspace alignment [208], correlation alignment [210], and iterative closest point (ICP) [214]. In both settings, HiWA exhibits strongest performance, with HiWA-SSC trailing closely behind (since clusters are independently resolved with SSC), followed by WA, then other algorithms. Subspace alignment methods have remarkably poor performance in higher dimensions due to their inability to resolve subspace sign ambiguities, while ICP demonstrates its notorious dependence on good initial conditions. These results indicates HiWA's strong robustness against initial conditions.

6.5 Alignment of neural coding datasets

Decoding intent (e.g., where you want to move your arm) or evoked responses (e.g., what you are looking at or listening to) directly from neural activity is a widely studied prob-

lem in neuroscience, and the first step in the design of a brain machine interface (BMI). A critical challenge in BMI is that neural decoders need to be recalibrated (or re-trained) due to drift in neural responses or electrophysiology measurements/readouts [215]. Recently, a method for semi-supervised brain decoding was proposed which finds a transformation between projected neural responses and movements by solving a KL-divergence minimization problem [216]. Using this approach, one could build robust decoders that work across days and shifts in neural responses through alignment.

To test the utility of hierarchical alignment for neural decoding, we utilized datasets collected from the primary motor cortex while a non-human primate (macaque monkey) was making arm movements during a center out reaching task [216]. After spike sorting and binning the data, we applied factor analysis to reduce the data dimensionality to 3D (source distribution) and then applied HiWA to align the neural data to a 3D movement distribution (target distribution) (Figure 6.3). We compared the performance of HiWA to a standard Wasserstein alignment (WA) that doesn't use a nested structure, and a baseline brute force search method called distribution alignment decoding (DAD) [216]. In all cases, we examined the accuracy in predicting the target reach direction for the motor decoding task at hand. This is akin to asking whether the algorithm predicted the correct cluster correspondences.

To examine the sensitivity of our method to quantities studied in our theory, we first examined the impact of the sampling density (Figure 6.3(b)) on performance. Surprisingly, HiWA continues to produce consistent cluster correspondences ($> 70\%$ accuracy), even as the number of samples per cluster drops to around 8 samples. In comparison, DAD is competitive for larger sample sizes but its performance rapidly drops off as sampling density decreases because it requires estimating a distribution. This is evidence that optimal transport's ability to exploit support geometry shines superior in comparison to DAD which requires a kernelized density estimation procedure to utilize the KL divergence. WA suffers from the presence of many local minima and fails to find the correct cluster corre-

spondences.

To study the impact of local and global geometry on whether an unlabeled source and target can be aligned, we applied HiWA to eight different subsets of reach directions (movement patterns). When just two reach directions are considered (Figure 6.3(c), Columns 1-4), global geometry becomes useless in determining the correct rotation. In this case, we observe that HiWA is only capable of consistently doing so when cluster asymmetries are sufficiently extreme in both the source and target to allow discernment. When three reach directions are considered (Figure 6.3(c), Columns 5-8), the global geometry can be used, yet there still exist symmetrical cases where recovering the correct rotation is unlikely without adequate local asymmetries or some supervised (labeled) data to match clusters. These results suggest that hierarchical structure can be critical in resolving ambiguities in alignment of globally symmetric movement distributions.

6.6 Discussion

This paper introduces a new method for hierarchical alignment with Wasserstein distances and provided an efficient numerical solution with analytic guarantees. We tested the method and compared its performance with other alignment methods on both synthetic mixture model datasets and in a neural decoding example. Our results on real neural datasets suggest that when either global or local cluster structure is preserved across datasets, a hierarchical approach can dramatically improve performance over traditional OT approaches.

While our approach demonstrates strong performance with unitary transformations, this could be restrictive when the data lives on more interesting topologies (e.g., structured manifolds [217]). Our hierarchical formulation could in principle provide the necessary structure to perform alignment over richer classes of transformations. Our results on neural data are compelling and suggest that HiWA can be applied to a wider range of neural datasets and higher dimensional alignment problems, such as aligning neural datasets across days without needing to match kinematics or another measured behavioral covariate.

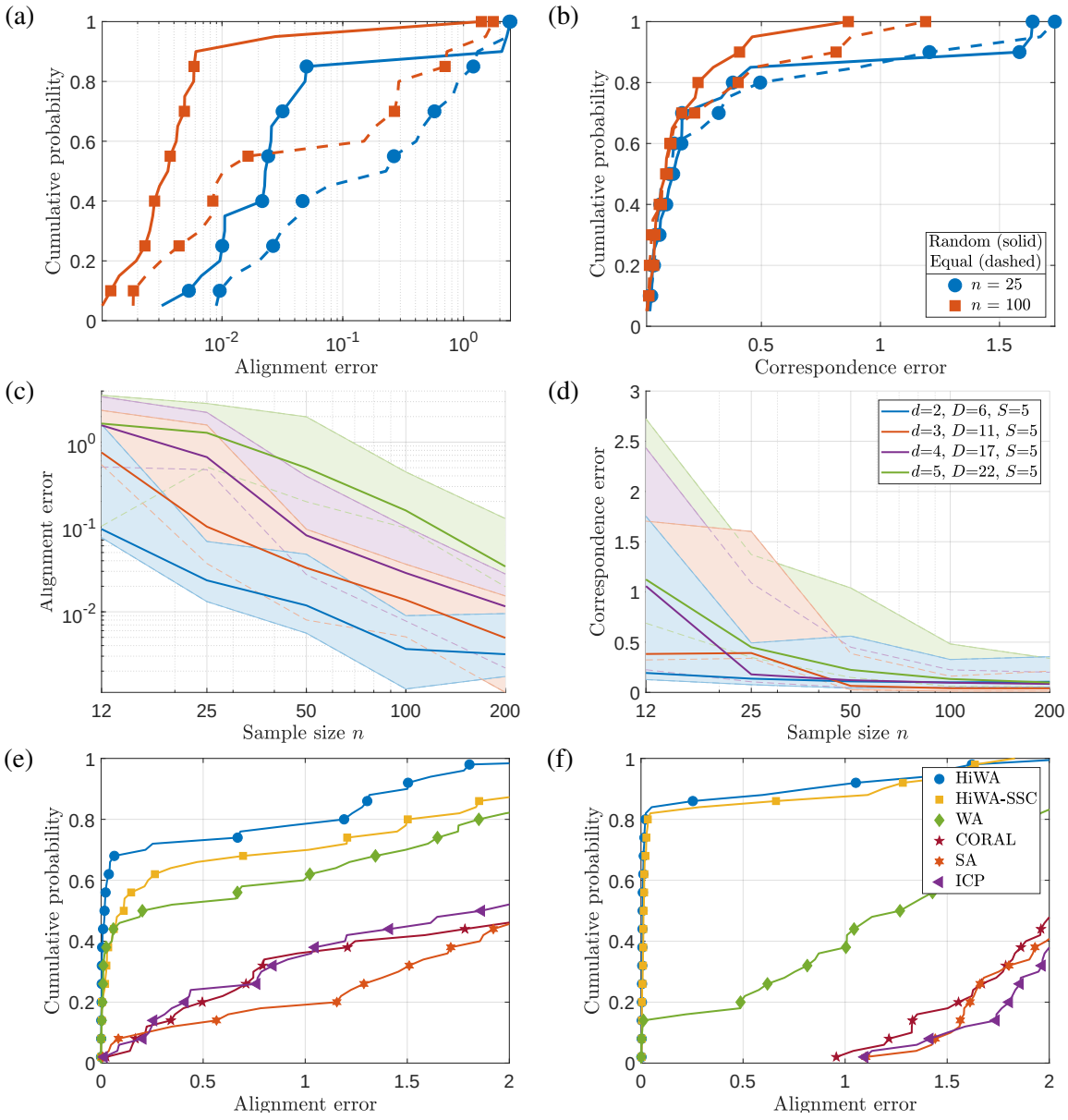


Figure 6.2: Synthetic experiments. HiWA was tested in two subspace configurations (a,b): randomly-spaced (average-case, solid) versus equally-spaced (worst-case, dashed) for $S = 5, d = 2, D = 6, n = \{25, 100\}$, where S is the number of clusters, d the dimension of each cluster, D is the embedding dimension, and n is the sample size. As expected, performance in terms of the (a) alignment and (b) correspondence (from 20 random trials) error is better in the average (vs. worst) case. In (c,d), we report (c) alignment and (d) correspondence errors as d and n varies, and report the error's $25^{\text{th}}/50^{\text{th}}/75^{\text{th}}$ percentiles from 20 trials. In (e,f), we compare HiWA when clusters are known (HiWA), HiWA when clusters are unknown (HiWA-SSC), non-hierarchical Wasserstein alignment (WA), subspace alignment methods (SA [208], CORAL [210]), and iterative closest point (ICP) [214] for $n = 50$, $d = 2$, and (e) $S = 5, D = 6$, and (f) $S = 2, D = 2$.

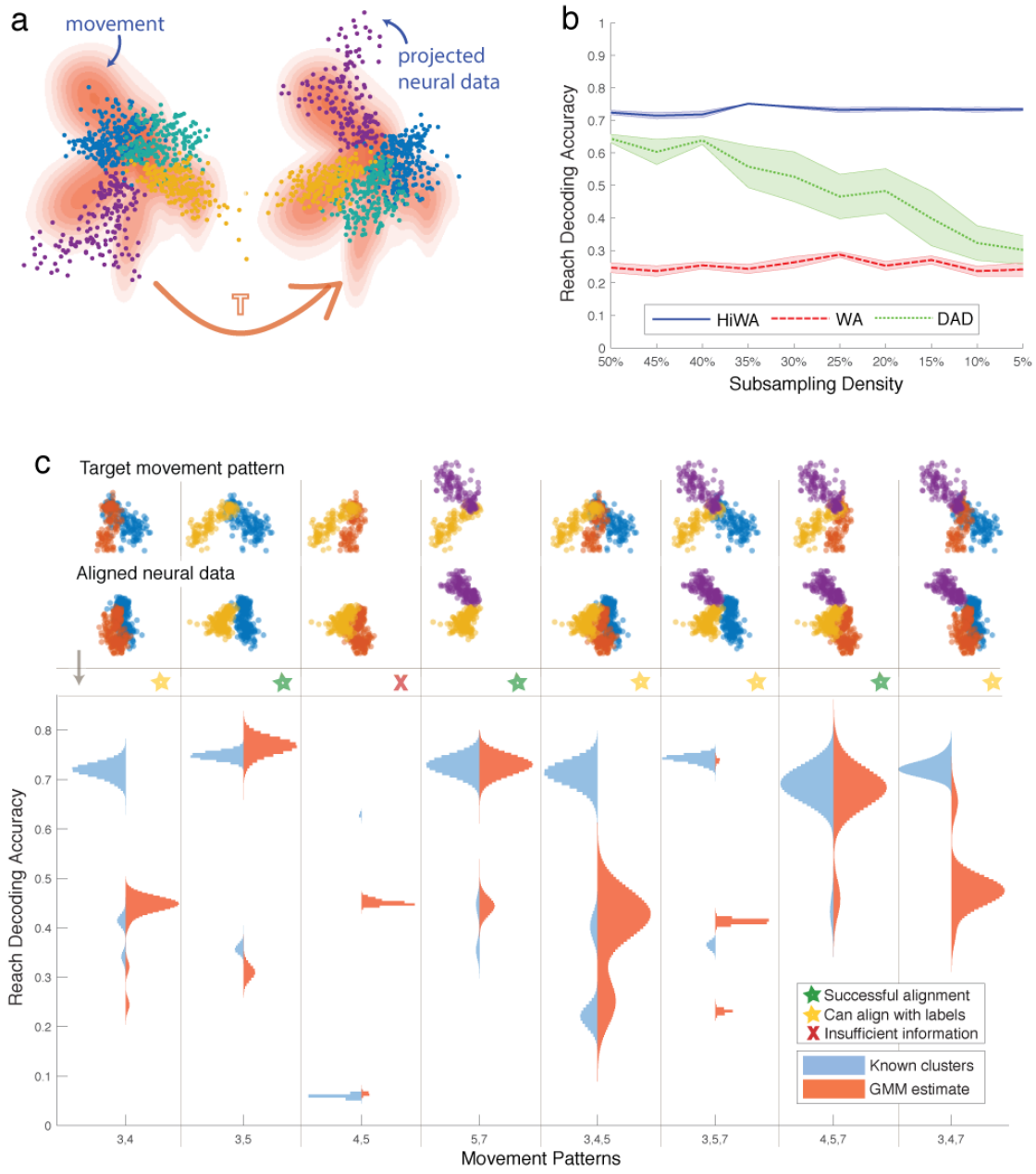


Figure 6.3: Results on brain decoding dataset. How distribution alignment is used to translate neural activity into movement – low-dimensional embeddings of neural data are aligned with target movement patterns (a). In (b), we compare the performance (cluster correspondence) of HiWA, WA, and DAD as the number of points in the source dataset decreases. Next, we compared the performance of HiWA with known and estimated clusters (via GMM). Movement patterns in which cluster separability is high and the geometry is preserved across datasets, can be aligned in both cases (green stars). Patterns where separability is low but geometry is useful can be aligned when the cluster arrangements are known (yellow stars), and when the geometry is not unique, it is not possible to find the correct alignment (red X).

CHAPTER 7
CONCLUSIONS AND FUTURE DIRECTIONS

Structure, when appropriately harnessed, enhances applications of engineering and machine learning. The contributions of this thesis explores aspects of geometric structures at two scales: the dynamics of the singular datum across time, and the alignability of population-level data.

- Chapter 3 expands upon a recent robust dynamical inference method (reweighted ℓ_1 dynamical filtering [148]) to develop a reweighted *total variation* method which propagates dynamics via second order *edge*-sparsity statistics. Since edge accuracy is of paramount interest in cell membrane tracking for patch clamp applications, this algorithm is extremely effective due to its unique ability to exaggerate edges. Additionally, the reweighted nature of this algorithm naturally synergizes with iterative first order solvers, thereby allowing parallel implementations with off-the-shelf hardware such as general purpose graphics processing units (GPGPUs).

- Chapters 4 and 5 introduce optimal transport regularizers for modeling mass transport phenomenology in inference problems. In order for these regularizers to be realized at scale in standard inference settings (e.g., imaging problems), this thesis addresses computational complexity issues that optimal transport programs are bottlenecked by. For optimal transport regularizers with *general* cost distances, we introduce a generic ADMM solver framework based on first order proximal Bregman divergences, which significantly speeds up the convergence rates (by $O(n \log n)$ over standard ADMM) by exploiting the respective gradients' geometries involved in the problem. In the special case when cost distances are Euclidean, a fluid-based reformulation takes advantage of such structure to significantly reduce the number of

optimization variables from $O(n^2)$ to $O(n)$, allowing us to tackle problems at scale by using standard proximal optimization methods.

- Chapter 6 introduces a new method for hierarchical alignment with Wasserstein distances, along with an accompanying distributed numerical solver. In addition to demonstrating the efficacy of our approach on synthetic and real datasets, we address fundamental questions of when a cluster-based alignment approach will work. Our theoretical results show that significantly improved alignment results indeed demand that distributions share very similar structures at local and global levels. Moreover, the Wasserstein distance, while geometrically attractive because it is kernel-free, has a sample complexity that is exponentially dependent on the intrinsic dimensions of the dataset.

A central theme in this thesis is the development of efficient tools that exploit geometric structure in data. While subsequent iterations of methods in this thesis can undoubtedly benefit from higher fidelity representations (e.g., higher ambient dimensions), these modifications come at tremendous algorithmic costs (e.g., the optimization size changes from $O(n^2)$ to $O(n^3)$). As examples, consider how

- the cell tracking applications (of Chapter 3) will significantly benefit from an additional Z-dimension to give a full description of the 3D cell membrane, or
- the unbalanced mass transport approach (of Chapters 4 and 5) become less of a critical feature when higher dimensions are used since mass will no longer traverse across unmodeled dimensions.

To mitigate the huge computational costs in future developments, significant efforts have been devoted in this thesis, to make algorithms scalable with distributed designs that exploit parallel hardware[1]. The results in this thesis therefore represents algorithmic stepping stones for future research. In particular, this work has opened two lines of future research:

[1]Incidentally, the cost of parallel GPU hardware is decreasing due to increasing mass market demands, e.g., cryptocurrency, proliferation of industrial applications of deep learning such as self-driving cars.

1. extensions of optimal transport regularizers, and

2. extensions of hierarchical alignment.

The establishment of fast tools for a large class of optimal transport regularizers facilitates the future development of new applications and models. Since the 1-Wasserstein regularizer (from Chapter 4) is particularly able to cope with problems at scale, new large scale application spaces now become viable (e.g., imaging or video); for example, one could consider the modeling of signal continuities across multiple data instantiations, or time-based 'inpainting' to fill-in missing entries in online settings with lossy datalinks. When geodesic spaces can be detailed, OT regularizers that allow generic costs distances (from Chapter 5) could introduce new applications; for example when functional magnetic resonance imaging (fMRI) and electroencephalogram (EEG) sensors are used in tandem, fMRI data (streaming in at higher fidelity but at a lower sampling rate than EEG) can be used to describe the cost distance of an optimal transport regularizer within an EEG's source localization framework (as described in Chapter 5) to more accurately locate brain activations in real-time. We believe that the Bregman ADMM framework (from Chapter 5) will facilitate the incorporation of new models (e.g., rigid shape models [218]) by virtue of ADMM's ability to accommodate a large (linear) class of optimal transport constraints/functions. While the application front looks promising, perhaps the most pressing theoretical work is to precisely describe how generic optimal transport quantifies displacement uncertainty, especially when used to assist at sparse recovery (e.g., LASSO) or low-rank signal separation (e.g., RPCA).

Our two-level hierarchical alignment framework (from Chapter 6) is a proof of concept that paves the way for many natural extensions. Extending this framework to accommodate deeper hierarchical structures for alignment (e.g., using convex hierarchical clustering approaches [219]) should in principle make the alignment even more robust towards local minima. The analysis framework developed in this work pioneers a path for evaluating the alignability of more general types of hierarchical structures. Finally, our work represents

a starting point for more complicated alignment frameworks, e.g., for jointly performing tasks like learning the latent representation and/or performing clustering. Despite the exciting plethora of algorithmic possibilities, a critical missing piece of analysis is required to shed light about the geometric relationship between the data's manifold structure and the alignment's transformation structure. This thesis conveniently skirts theses issues by assuming subspace structure in the data and an isometric alignment transformation; yet it is crucial to more generally understand what types of alignment transformations preserve general notions of manifold structure in the data and the associated alignability conditions.

Appendices

APPENDIX A
CONVERGENCE OF UNBALANCED PRIMAL-DUAL PROXIMAL ALGORITHM

Proof. We proceed by showing that the conditions of [24, Theorem 1] are satisfied. First, note that we may rewrite (4.8) as

$$\mathcal{L}(\boldsymbol{M}, \boldsymbol{r}, \boldsymbol{x}, \boldsymbol{a}) = G(\boldsymbol{M}, \boldsymbol{r}, \boldsymbol{x}) + \langle \boldsymbol{a}, \boldsymbol{K}\boldsymbol{b} \rangle - F(\boldsymbol{a}),$$

where $G(\boldsymbol{M}, \boldsymbol{r}, \boldsymbol{x}) = \|\boldsymbol{M}\|_{2,1} + \mu \|\boldsymbol{r}\|_p^p + \frac{1}{2\rho} \|\boldsymbol{x} - \boldsymbol{p}\|_2^2 + \iota_+(\boldsymbol{x})$, $\boldsymbol{K} = [\boldsymbol{D}, -\boldsymbol{I}, \boldsymbol{I}, -\boldsymbol{I}]$, $\boldsymbol{D}$ is the matrix corresponding to the divergence operator, $\boldsymbol{b} = [\mathrm{vec}(\boldsymbol{M}); \boldsymbol{x}; \boldsymbol{r}]$ and $F(\boldsymbol{a}) = 0$. The functions G and F are proper, convex, and lower semi-continuous, and K is a linear operator. Note that if λ is an eigenvalue of a matrix $\boldsymbol{B}^*\boldsymbol{B}$, then $\lambda + 1$ is an eigenvalue of the matrix $\widetilde{\boldsymbol{B}}^*\widetilde{\boldsymbol{B}}$ where $\widetilde{\boldsymbol{B}} = [\boldsymbol{B}, \pm\boldsymbol{I}]$ (this is easily verified by noting that if $\boldsymbol{v}$ is the corresponding eigenvector of $\boldsymbol{B}^*\boldsymbol{B}$, then $[\lambda\boldsymbol{v}; \pm\boldsymbol{B}\boldsymbol{v}]$ is the corresponding eigenvector of $\widetilde{\boldsymbol{B}}^*\widetilde{\boldsymbol{B}}$). By repeated application of this identity, it follows that $\lambda_{\max}(\boldsymbol{K}) = \lambda_{\max}(\boldsymbol{D}^*\boldsymbol{D}) + 3$. Since $\boldsymbol{D}\boldsymbol{D}^*$ is the discrete Laplacian operator, $\lambda_{\max}(\nabla^2) + 3$ is the maximum eigenvalue of $\boldsymbol{K}$. Thus, the conditions of Theorem 1 of Chambolle and Pock are satisfied when $\tau_1\tau_2 < \frac{1}{\lambda_{\max}(\nabla^2)+3}$. □

APPENDIX B

PROOFS FOR KL-EUCLIDEAN BREGMAN ADMM ALGORITHM

B.1 Convergence guarantee with a composite Bregman divergence

In the following, we demonstrate convergence of the proposed ADMM iterations (5.5)-(5.7) by adapting the convergence proof in [36] to the proposed composite Bregman divergence (5.4). To simply our analysis, we let all affine operators $\mathcal{A}, \mathcal{B}$ be identity. We first simplify notation by concatenating variables in the following fashion:

$$\vec{x} = (\boldsymbol{x}, \boldsymbol{P}), \quad \vec{z} = (\boldsymbol{z}, \boldsymbol{R}), \quad \vec{a} = (\boldsymbol{a}, \boldsymbol{B}).$$

Under this notation, we re-express (5.3) as

$$\min_{\vec{x},\vec{z}} \; c(\vec{x}) + h(\vec{z}) \quad \text{s.t.} \quad \vec{x} = \vec{z} \tag{B.1}$$

where $c(\vec{x}) \equiv f(\boldsymbol{x}, \boldsymbol{P}) + \iota_+(\boldsymbol{x}, \boldsymbol{P}) + \kappa \langle \boldsymbol{P}, \boldsymbol{C} \rangle$, and $h(\vec{z}) \equiv g(\boldsymbol{z}, \boldsymbol{R})$.

We make the following assumption hereon:

Assumption B.1.1. *$c : (\mathbb{R}^n, \mathbb{R}^{n\times n}) \mapsto \mathbb{R} \cup \{+\infty\}$ and $h : (\mathbb{R}^n, \mathbb{R}^{n\times n}) \mapsto \mathbb{R} \cup \{+\infty\}$ are closed, proper and convex, and an optimal solution exists.*

Definition B.1.2 (Composite Bregman divergence)**.** *Let the composite Bregman divergence and its gradient w.r.t. its first argument be*

$$B(\vec{x}|\vec{z}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + KL(\boldsymbol{P}|\boldsymbol{Q})$$

$$\Rightarrow \quad \nabla_{\vec{x}} B(\vec{x}|\vec{z}) = (\boldsymbol{x} - \boldsymbol{y}, \log(\boldsymbol{P}) - \log(\boldsymbol{Q})) \,.$$

The subsequent Bregman ADMM steps are equivalently stated as

$$\begin{aligned}
\vec{x}^{(t+1)} &\leftarrow \arg\min_{\vec{x}} \; c(\vec{x}) + \langle \vec{a}^{(t)}, \vec{x} \rangle + \rho B(\vec{x}|\vec{z}^{(t)}), \\
\vec{z}^{(t+1)} &\leftarrow \arg\min_{\vec{z}} \; h(\vec{z}) + \langle \vec{a}^{(t)}, -\vec{z} \rangle + \rho B(\vec{z}|\vec{x}^{(t+1)}), \\
\vec{a}^{(t+1)} &\leftarrow \vec{a}^{(t)} + \tau(\vec{x}^{(t+1)} - \vec{z}^{(t+1)}),
\end{aligned} \tag{B.2}$$

The (un-augmented) Lagrangian of (B.1) is

$$\mathcal{L}(\vec{x}, \vec{z}, \vec{a}) = c(\vec{x}) + h(\vec{z}) + \langle \vec{a}, \vec{x} - \vec{z} \rangle \, . \tag{B.3}$$

The Lagrangian's optimal solution $\{\vec{x}^\star, \vec{z}^\star, \vec{a}^\star\}$ has to satisfy the following KKT conditions:

$$-\vec{a}^\star \in \partial c(\vec{x}^\star), \quad \vec{a}^\star \in \partial h(\vec{z}^\star), \quad \vec{x}^\star = \vec{z}^\star. \tag{B.4}$$

The first order stationarity conditions of the first two ADMM updates in (B.2) are

$$\begin{aligned}
-\vec{a}^{(t)} - \rho \nabla_{\vec{x}} B(\vec{x}^{(t+1)}|\vec{z}^{(t)}) &\in \partial c(\vec{x}^{(t+1)}), \\
\vec{a}^{(t)} - \rho \nabla_{\vec{z}} B(\vec{z}^{(t+1)}|\vec{x}^{(t+1)}) &\in \partial h(\vec{z}^{(t+1)}),
\end{aligned} \tag{B.5}$$

which yields KKT optimality conditions (B.4) when the following are met:

$$\vec{x}^{(t+1)} = \vec{z}^{(t)}, \quad \vec{x}^{(t+1)} = \vec{z}^{(t+1)}. \tag{B.6}$$

Definition B.1.3 (Residual term)**.** *Let a residual term be defined with parameter $\gamma > 0$ as*

$$r^{(t+1)} = B(\vec{x}^{(t+1)}|\vec{z}^{(t)}) + \gamma \|\vec{x}^{(t+1)} - \vec{z}^{(t+1)}\|_2^2. \tag{B.7}$$

To show convergence of the algorithm (i.e., that (B.4) and (B.5) are satisfied), it is sufficient to show residual convergence (i.e., $r^{(t+1)} = 0$), which is a Lyapunov function for the algorithm, i.e., a nonnegative scalar that decreases in each iteration. We exploit the

convergence result (theorem 1) from [36] whose prerequisite is α-strong convexity on the chosen Bregman divergence. We demonstrate this with the following proposition.

Proposition B.1.4. *The Bregman divergence (from definition B.1.2) is α-strongly convex*

$$B(\vec{x}|\vec{z}) \geq \frac{\alpha}{2}\Big(\|\boldsymbol{x}-\boldsymbol{z}\|_p^2 + \|\operatorname{vec}(\boldsymbol{P}-\boldsymbol{R})\|_p^2\Big),$$

for $p=1$ and $\alpha = \min(\frac{1}{c_1}, \frac{1}{c_2})$, with $c_1 = \dim(\boldsymbol{x}), c_2 = \mathbb{1}^\top \boldsymbol{P}\mathbb{1}$.

Proof. We construct this inequality in two parts. For the first part, we have

$$\|\boldsymbol{x}-\boldsymbol{z}\|_2 \geq \frac{1}{\sqrt{c_1}}\|\boldsymbol{x}-\boldsymbol{z}\|_1 \Leftrightarrow \|\boldsymbol{x}-\boldsymbol{z}\|_2^2 \geq \frac{1}{c_1}\|\boldsymbol{x}-\boldsymbol{z}\|_1^2, \tag{B.8}$$

as a direct consequence of Cauchy-Schwarz, and where $c_1 = \dim(\boldsymbol{x}) = \dim(\boldsymbol{z})$.

For the second part, we need to scale the strong convexity property of the KL-divergence w.r.t. ℓ_1 norm over the unit simplex, i.e., $\mathrm{KL}(\boldsymbol{p}|\boldsymbol{r}) \geq \frac{1}{2}\|\boldsymbol{p}-\boldsymbol{r}\|_1^2$ [35]. We define $\boldsymbol{p}' = \operatorname{vec}(\boldsymbol{P}), \boldsymbol{r}' = \operatorname{vec}(\boldsymbol{R})$ and $c_2 = \mathbb{1}^\top \boldsymbol{p}' = \mathbb{1}^\top \boldsymbol{r}'$; these are related to their unit simplex counterparts $\boldsymbol{p}, \boldsymbol{r}$ via the relation $\boldsymbol{p}' = c_2\boldsymbol{p}, \boldsymbol{r}' = c_2\boldsymbol{r}$. The *scaled* strong convexity property for the KL-divergence is derived as:

$$\begin{aligned}
\mathrm{KL}(\boldsymbol{p}'|\boldsymbol{r}') &= \sum_i p_i' \log \frac{p_i'}{r_i'} - p_i' + r_i' \\
&= c_2 \sum_i p_i \log \frac{p_i}{r_i} - p_i + r_i = c_2 \mathrm{KL}(\boldsymbol{p}|\boldsymbol{r}) \\
&\geq \frac{c_2}{2}\|\boldsymbol{p}-\boldsymbol{r}\|_1^2 = \frac{c_2}{2}\|\boldsymbol{p}'/c_2 - \boldsymbol{r}'/c_2\|_1^2 \\
&= \frac{1}{2c_2}\|\boldsymbol{p}'-\boldsymbol{r}'\|_1^2.
\end{aligned}$$

Putting the two parts together, we have

$$\begin{aligned} B(\vec{x}|\vec{z}) &= \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2 + \mathrm{KL}(\boldsymbol{P}|\boldsymbol{R}) \\ &\geq \frac{1}{2c_1}\|\boldsymbol{x}-\boldsymbol{z}\|_1^2 + \frac{1}{2c_2}\|\operatorname{vec}(\boldsymbol{P}-\boldsymbol{R})\|_1^2 \\ &\geq \frac{\min(1/c_1, 1/c_2)}{2}\Big(\|\boldsymbol{x}-\boldsymbol{z}\|_1^2 + \|\operatorname{vec}(\boldsymbol{P}-\boldsymbol{R})\|_1^2\Big). \end{aligned}$$

□

Corollary B.1.5 (Composite Bregman ADMM convergence)**.** *Let the sequence* $\{\vec{x}^{(t+1)}, \vec{z}^{(t+1)}, \vec{a}^{(t+1)}\}$ *be generated by Bregman ADMM updates* (B.2)*, and* $\{\vec{x}^\star, \vec{z}^\star, \vec{a}^\star\}$ *satisfies the KKT conditions* (B.4)*. Let assumption B.1.1 hold, and* $\tau \leq \rho\alpha - 2\gamma$ *and* $0 < \gamma < \frac{\rho\alpha}{2}$*, where* $p = 1$ *and* $\alpha = \min(\frac{1}{c_1}, \frac{1}{c_2})$ *with* $c_1 = \dim(\boldsymbol{x})$ *and* $c_2 = \mathbb{1}^\top \boldsymbol{P}\mathbb{1}$*. Then* $r^{(t+1)}$ *as given in Definition B.1.3, converges to zero and the sequence* $\{\vec{x}^{(t+1)}, \vec{z}^{(t+1)}, \vec{a}^{(t+1)}\}$ *converges to a KKT point* $\{\vec{x}^\star, \vec{z}^\star, \vec{a}^\star\}$ *of the problem's Lagrangian* (B.3)*.*

Proof. Based on Proposition B.1.4, $B(\cdot|\cdot)$ is α-strongly convex according to the $p = 1$ norm, with $\alpha = \min(\frac{1}{c_1}, \frac{1}{c_2})$ where $c_1 = \dim(\boldsymbol{x})$ and $c_2 = \mathbb{1}^\top \boldsymbol{P}\mathbb{1}$. As a consequence, lemma 2 in [36] applies, and it returns bounds $\tau \leq \rho\alpha - 2\gamma$ and $0 < \gamma < \frac{\rho\alpha}{2}$. Finally, theorem 1 in [36] establishes convergence by demonstrating convergence of residual $r^{(t+1)} \to 0$ and therefore $\{\vec{x}^{(t+1)}, \vec{z}^{(t+1)}, \vec{a}^{(t+1)}\}$ to a KKT point $\{\vec{x}^\star, \vec{z}^\star, \vec{a}^\star\}$. □

B.2 Proofs of proximal KL-Euclidean algorithms

B.2.1 Proof of Theorem 5.1.3

Proof. Applying Lagrangian multipliers $\boldsymbol{\lambda}$ to account for the equality constraint in (5.10) yields the minimax point problem

$$\min_{\boldsymbol{z},\boldsymbol{R}} \max_{\boldsymbol{\lambda}} \; \langle \boldsymbol{a}, -\boldsymbol{z} \rangle + \langle \boldsymbol{B}, -\boldsymbol{R} \rangle \\ + \rho B(\boldsymbol{z}, \boldsymbol{R} | \boldsymbol{x}, \boldsymbol{P}) + \langle \boldsymbol{\lambda}, \boldsymbol{R}\mathbb{1} - \boldsymbol{z} \rangle \,,$$

where B is defined in (5.4). The stationary point $\{\boldsymbol{z}^\star, \boldsymbol{R}^\star, \boldsymbol{\lambda}^\star\}$ satisfies the following (element-wise) KKT conditions

$$\begin{aligned} \lambda_i^\star &= -a_i + \rho(z_i^\star - x_i), \\ \lambda_i^\star &= B_{ij} - \rho \log(R_{ij}^\star / P_{ij}), \\ \sum_j R_{ij}^\star &= z_i^\star. \end{aligned}$$

Eliminating $\lambda_i^\star$ by equating the first two KKT conditions yields the relation $R_{ij}^\star \exp(z_i^\star) = S_{ij}$, where $S_{ij} = \exp((a_i + B_{ij})/\rho + x_i + \log(P_{ij}))$. Combining this relation with the third KKT condition results in the stated closed form solutions. □

B.2.2 Proof of Theorem 5.1.4

Proof. Applying Lagrangian multipliers $\boldsymbol{\lambda}$ to account for the equality constraint in (5.11) yields the minimax problem

$$\min_{\boldsymbol{u},\boldsymbol{z},\boldsymbol{R}\geq 0} \max_{\boldsymbol{\lambda}} \; \frac{\mu}{2} \|\boldsymbol{u}\|_2^2 + \langle \boldsymbol{a}, -\boldsymbol{z} \rangle + \langle \boldsymbol{B}, -\boldsymbol{R} \rangle \\ + \rho B(\boldsymbol{z}, \boldsymbol{R} | \boldsymbol{x}, \boldsymbol{P}) + \langle \boldsymbol{\lambda}, \boldsymbol{R}\mathbb{1} - \boldsymbol{z} - \boldsymbol{u} \rangle \,,$$

where B is defined in (5.4). We begin by computing the *unconstrained* solution of the minimax problem to check which non-negativity constrains are violated. The stationary point $\{\widehat{\boldsymbol{z}}^\star, \widehat{\boldsymbol{R}}^\star, \widehat{\boldsymbol{u}}^\star, \widehat{\boldsymbol{\lambda}}^\star\}$ satisfies the following (element-wise) KKT conditions

$$\begin{aligned}
\widehat{\lambda}_i^\star &= \mu \widehat{u}_i^\star, \\
\widehat{\lambda}_i^\star &= -a_i + \rho(\widehat{z}_i^\star - x_i), \\
\widehat{\lambda}_i^\star &= B_{ij} - \rho \log(\widehat{R}_{ij}^\star / P_{ij}), \\
\sum_j \widehat{R}_{ij}^\star &= \widehat{z}_i^\star + \widehat{u}_i^\star,
\end{aligned}$$

where the first three were derived from stationarity conditions. To eliminate $\widehat{u}_i^\star$ and $\widehat{\lambda}_i^\star$, we combine the first, second and fourth KKT conditions to yield $\sum_j \widehat{R}_{ij}^\star = \alpha \widehat{z}_i^\star - p_i$, where $\alpha = 1 + \rho/\mu$ and $p_i = (a_i + \rho x_i)/\mu$. Combining the second and third KKT conditions yields $\widehat{R}_{ij}^\star \exp(\widehat{z}_i^\star) = S_{ij}$, where $S_{ij} = \exp(B_{ij}/\rho + \log P_{ij} + (a_i/\rho + x_i))$. Equating the two derived relations yields $\alpha \widehat{z}_i^\star \exp \widehat{z}_i^\star - p_i \exp \widehat{z}_i^\star = \sum_j S_{ij}$, which has the solution $\widehat{z}_i^\star = \omega((\exp(-p_i/\alpha) \sum_j S_{ij})/\alpha) + p_i/\alpha$, where ω refers to the Lambert W function. We solve for $\widehat{R}_{ij}^\star$ by the second derived relation, i.e., $\widehat{R}_{ij}^\star = S_{ij}/\exp(\widehat{z}_i^\star)$, and accordingly $\widehat{u}_i^\star = \sum_j \widehat{R}_{ij}^\star - \widehat{z}_i^\star$.

Next, we solve the original *non-negativity-constrained* problem. We first delineate the three cases of the problem and proceed to tackle them individually: (i) $z_i, u_i > 0$, (ii) $z_i \geq 0, u_i = 0$, (iii) $z_i = 0, u_i \geq 0$. In the first case, element-indices are identified using the set $\Omega_1 := \{i \in [\![n]\!] : \widehat{z}_i, \widehat{u}_i > 0\}$. For elements in this set, solutions are identical to the unconstrained problem, i.e., $z_i^\star = \widehat{z}_i^\star, u_i^\star = \widehat{u}_i^\star, R_{ij}^\star = \widehat{R}_{ij}^\star, \forall i \in \Omega_1$. In the second case, element-indices are identified using the set $\Omega_2 := \{i \in [\![n]\!] : \widehat{u}_i \leq 0\}$. Since $u_i = 0$, this allows us to reduce the problem to that of (5.10). It therefore follows the solution $z_i^\star = \omega(\sum_j S_{ij})$, and $R_{ij}^\star = \exp(-z_i^\star) S_{ij}, \forall i \in \Omega_2$, where $S_{ij} = \exp(\log P_{ij} + B_{ij}/\rho + a_i/\rho + x_i)$, and ω refers to the Lambert W function. In the third case, element-indices are identified using the set $\Omega_3 := \{i \in [\![n]\!] : \widehat{z}_i \leq 0\}$. In this case, we need to re-solve the problem with

$z_i = 0$. The KKT conditions are now:

$$\lambda_i^\star = \mu u_i^\star, \qquad \lambda_i^\star = B_{ij} - \rho \log(R_{ij}^\star / P_{ij}), \qquad \sum_j R_{ij}^\star = u_i^\star.$$

By combining the first two conditions, we have $R_{ij}^\star = Q_{ij} \exp(-\beta u_i^\star)$, where $Q_{ij} = \exp(\log P_{ij} + B_{ij}/\rho)$, and $\beta = \mu/\rho$. Combining this with the last condition yields the solution for $u_i^\star = \omega(\beta \sum_j Q_{ij})/\beta$. □

B.2.3 Proof of Theorem 5.1.5

Proof. We consider the equivalent problem that replaces the $\|\boldsymbol{u}\|_1$ with $\langle \mathbb{1}, \boldsymbol{u} \rangle$ since $\boldsymbol{u} \geq 0$. Applying Lagrangian multipliers $\boldsymbol{\lambda}$ to account for the equality constraint in (5.12) yields the minimax problem

$$\min_{\boldsymbol{u},\boldsymbol{z},\boldsymbol{R}\geq 0} \max_{\boldsymbol{\lambda}} \; \mu \langle \mathbb{1}, \boldsymbol{u} \rangle + \langle \boldsymbol{a}, -\boldsymbol{z} \rangle + \langle \boldsymbol{B}, -\boldsymbol{R} \rangle \\ + \rho B(\boldsymbol{z}, \boldsymbol{R} | \boldsymbol{x}, \boldsymbol{P}) + \langle \boldsymbol{\lambda}, \boldsymbol{R}\mathbb{1} - \boldsymbol{z} - \boldsymbol{u} \rangle ,$$

where B is defined in (5.4). We begin by computing the *unconstrained* solution of the minimax problem to check which non-negativity constrains are violated. The stationary point $\{\widehat{\boldsymbol{z}}^\star, \widehat{\boldsymbol{R}}^\star, \widehat{\boldsymbol{u}}^\star, \widehat{\boldsymbol{\lambda}}^\star\}$ will satisfy the following (element-wise) KKT conditions:

$$\begin{aligned} \widehat{\lambda}_i^\star &= \mu, \\ \widehat{\lambda}_i^\star &= -a_i + \rho(\widehat{z}_i^\star - x_i), \\ \widehat{\lambda}_i^\star &= B_{ij} - \rho \log(\widehat{R}_{ij}^\star / P_{ij}), \\ \sum_j \widehat{R}_{ij}^\star &= \widehat{z}_i^\star + \widehat{u}_i^\star, \end{aligned}$$

where the first three were derived from stationarity conditions. By combining select KKT conditions, we obtain closed form relations: $\widehat{z}_i = (\mu + a_i)/\rho + x_i$ via the first two conditions,

and $\widehat{R}_{ij} = \exp(\log P_{ij} + B_{ij}/\rho - \mu/\rho)$ via the first and third conditions. These two relations are combined with the fourth KKT condition to yield $\widehat{u}_i = \sum_j \widehat{R}_{ij} - \widehat{z}_i$.

Next, we will consider two cases to the original non-negativity-constrained problem: (i) $u_i > 0$, (ii) $u_i = 0$. We first identify indices belonging to each case by defining the set $\Omega := \{i \in [\![n]\!] : \widehat{u}_i > 0\}$ (which corresponds to the first case). In the first case, R_{ij} is separable from z_i so we allow $R^\star_{ij} = \widehat{R}_{ij}, \forall i \in \Omega$. Similarly, z_i's separability from R_{ij} and its quadratic objective allows us to simplify its solution to a mere projection, i.e., $z^\star_i = \max(0, \widehat{z}_i), \forall i \in \Omega$. In the second case, we find that $u_i = 0$ allows us to reduce the problem to that of (5.10). It therefore follows the solution $z^\star_i = \omega(\sum_j S_{ij})$, and $R^\star_{ij} = \exp(-z^\star_i) S_{ij}$, $\forall i \in \bar{\Omega}$ (complement of Ω), where $S_{ij} = \exp(\log P_{ij} + B_{ij}/\rho + a_i/\rho + x_i)$, and ω refers to the Lambert W function. □

B.3 Derivation of ADMM algorithms in Section 5.2

For Bregman ADMM or (standard) ADMM, the update steps are:

$$
\begin{aligned}
\boldsymbol{x}^{(t+1)} \leftarrow \operatorname*{arg\,min}_{\boldsymbol{x} \geq 0} \;& \left\langle \boldsymbol{a}_1^{(t)}, \boldsymbol{x} \right\rangle + \frac{\rho}{2} \|\boldsymbol{x} - \boldsymbol{z}_1^{(t)}\|_2^2 \\
& + \left\langle \boldsymbol{a}_2^{(t)}, \boldsymbol{x} \right\rangle + \frac{\rho}{2} \|\boldsymbol{x} - \boldsymbol{z}_2^{(t)}\|_2^2, \\
\boldsymbol{P}^{(t+1)} \leftarrow \operatorname*{arg\,min}_{\boldsymbol{P}^\top \mathbb{1} = \widehat{\boldsymbol{x}}} \;& \iota_+(\boldsymbol{P}) + \left\langle \boldsymbol{P}, \kappa \boldsymbol{C} \right\rangle \\
& + \left\langle \boldsymbol{P}, \boldsymbol{B}_1^{(t)} \right\rangle + \rho \mathrm{KL}(\boldsymbol{P} | \boldsymbol{R}_1^{(t)}), \\
(\boldsymbol{z}_1^{(t+1)}, \boldsymbol{R}_1^{(t+1)}) \leftarrow \operatorname*{arg\,min}_{\boldsymbol{R}_1 \mathbb{1} = \boldsymbol{z}_1} \;& - \left\langle \boldsymbol{a}_1^{(t)}, \boldsymbol{z}_1 \right\rangle + \frac{\rho}{2} \|\boldsymbol{z}_1 - \boldsymbol{x}^{(t+1)}\|_2^2 \\
& - \left\langle \boldsymbol{B}_1^{(t)}, \boldsymbol{R}_1 \right\rangle + \rho \mathrm{KL}(\boldsymbol{R}_1 | \boldsymbol{P}^{(t+1)}), \\
\boldsymbol{z}_2^{(t+1)} \leftarrow \operatorname*{arg\,min}_{\boldsymbol{z}_2} \;& \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{\Phi} \boldsymbol{z}_2\|_2^2 \\
& + \left\langle \boldsymbol{a}_2^{(t)}, -\boldsymbol{z}_2 \right\rangle + \frac{\rho}{2} \|\boldsymbol{z}_2 - \boldsymbol{x}^{(t+1)}\|_2^2,
\end{aligned}
$$

$$\boldsymbol{a}_1^{(t+1)} \leftarrow \boldsymbol{a}_1^{(t)} + \tau(\boldsymbol{x}_1^{(t+1)} - \boldsymbol{z}_1^{(t+1)}),$$
$$\boldsymbol{a}_2^{(t+1)} \leftarrow \boldsymbol{a}_2^{(t)} + \tau(\boldsymbol{x}^{(t+1)} - \boldsymbol{z}_2^{(t+1)}),$$
$$\boldsymbol{B}_1^{(t+1)} \leftarrow \boldsymbol{B}_1^{(t)} + \tau(\boldsymbol{P}^{(t+1)} - \boldsymbol{R}_1^{(t+1)}).$$

The $\boldsymbol{P}$ updates are closed-form exponentiated gradient steps [36] in Bregman ADMM, and a simplex projection algorithm [163] in standard ADMM. The $(\boldsymbol{z}_1, \boldsymbol{R}_1)$ updates are outlined in Section 5.1.4 for Bregman ADMM, and for standard ADMM, they are $\boldsymbol{z}_1 = ((\boldsymbol{B}_1/\rho + \boldsymbol{P})\mathbb{1} + n(\boldsymbol{a}_1/\rho + \boldsymbol{x}))/(n+1)$, $\boldsymbol{R}_1 = \boldsymbol{B}_1/\rho + \boldsymbol{P} + (\boldsymbol{a}/\rho + \boldsymbol{x} - \boldsymbol{z}_1)\mathbb{1}^\top$ (derived using KKT conditions).

For Sinkhorn ADMM, the update steps for $\boldsymbol{x}, \boldsymbol{z}_2, \boldsymbol{a}_1, \boldsymbol{a}_2$ are similar. The key difference is that there are no coupling variables (i.e., $\boldsymbol{P}, \boldsymbol{R}_1, \boldsymbol{B}_1$) since these are baked into the proximal algorithm. The $\boldsymbol{z}_1$ update is now:

$$\boldsymbol{z}_1^{(t+1)} \leftarrow \underset{\boldsymbol{z}_1}{\arg\min}\ \kappa\widehat{W}(\boldsymbol{z}_1, \widehat{\boldsymbol{x}}) + \langle \boldsymbol{b}^{(t)}, -\boldsymbol{z}_1 \rangle + \frac{\rho}{2}\|\boldsymbol{z}_1 - \boldsymbol{x}^{(t+1)}\|_2^2,$$

where $\widehat{W}$ refers to the approximated balanced OT problem, and whose update may be computed by the generalized Sinkhorn proximal algorithm [46].

APPENDIX C

PROOFS FOR ANALYSIS OF HIERARCHICAL OT ALIGNMENT

C.1 Alignability: existence conditions for unique cluster-correspondence

C.1.1 Part 1: Geometric perturbations conditions over the Birkhoff polytope

First, we provide the following lemma that illuminates some basic geometrical insights of the general OT cost matrix, whose coupling is in the Birkhoff polytope $\mathcal{B}_c := \{\boldsymbol{P} \in \mathbb{R}^{c\times c} : \boldsymbol{P}\mathbb{1}_c = \boldsymbol{P}^\top \mathbb{1}_c = \mathbb{1}_c\}$. We define the OT program with respect to a cost matrix $\boldsymbol{C}$ as

$$L(\boldsymbol{C}) := \underset{\boldsymbol{P}\in\mathcal{B}_c}{\arg\min} \ \langle \boldsymbol{P}, \boldsymbol{C}\rangle \,. \tag{C.1}$$

The following lemma describes the conditions on $\boldsymbol{C}$ for $\boldsymbol{P}$ to remain unperturbed at $\boldsymbol{P}^\star$.

Lemma C.1.1. *Define the set of vertices on the c-th Birkhoff polytope $\mathcal{B}_c$ that are within a δ-ball from $\boldsymbol{P}^\star$ as*

$$\mathcal{P}_\delta(\boldsymbol{P}^\star) = \{\boldsymbol{P} \in \mathcal{B}_c \setminus \{\boldsymbol{P}^\star\} : \|\boldsymbol{P} - \boldsymbol{P}^\star\|_F \le \delta\}. \tag{C.2}$$

Define the set of matrices that denote directions from $\boldsymbol{P}^\star$ to each neighboring vertex $\boldsymbol{P}_i$ as

$$\mathcal{V}_\delta(\boldsymbol{P}^\star) = \{\boldsymbol{V} \in \mathbb{R}^{c\times c} : \boldsymbol{V} = \boldsymbol{P} - \boldsymbol{P}^\star, \ \ \boldsymbol{P} \in \mathcal{P}_\delta(\boldsymbol{P}^\star)\}. \tag{C.3}$$

For the linear program's solution $\boldsymbol{P}^\star = L(\boldsymbol{C})$ to remain unchanged, $\forall \boldsymbol{V} \in \mathcal{V}_\delta(\boldsymbol{P}^\star)$ for $\delta = 2$, the cost matrix $\boldsymbol{C}$ should satisfy

$$\langle \boldsymbol{C}, \boldsymbol{V}\rangle > 0. \tag{C.4}$$

Proof. Birkhoff-von Neuman's theorem [220] states that the optimal transport solution must lie on the convex hull of the c-th Birkhoff polytope $\mathcal{B}_c$, and that its vertices are in fact permutation matrices. We therefore say that an LP solution $\boldsymbol{P}^\star$ is a vertex on $\mathcal{B}_c$. The outline of this proof is straightforward: so that $\boldsymbol{P}^\star$ remains unchanged, $\boldsymbol{C}$ should not cause $\boldsymbol{P}^\star$ to move to an adjacent edge of the Birkhoff polytope, nor should it cause it to extend beyond its adjacent edge because then, the neighboring vertex would assume the new solution. For the rest of this proof, we shall let $\boldsymbol{P}^\star = \boldsymbol{I}$ without any loss of generality. We define the set of nearest neighbors to $\boldsymbol{P}^\star$, which are simply permutation matrices that can be described as taking $\boldsymbol{P}^\star$ and exchanging any two columns. For a $\boldsymbol{P}$ with any two columns of $\boldsymbol{P}^\star$ exchanged, notice that the difference matrix $\boldsymbol{V} = \boldsymbol{P} - \boldsymbol{P}^\star$ is a symmetric matrix of mostly zeros except for two off-diagonal $+1$ entries and two diagonal -1 entries, hence $\|\boldsymbol{P} - \boldsymbol{P}^\star\|_F = 2$. Formally, we describe the set of nearest neighbors with $\mathcal{P}_\delta(\boldsymbol{P}^\star)$ with $\delta = 2$, defined by (C.2). Next, we define edges that are adjacent to $\boldsymbol{P}^\star$ using the set $\mathcal{V}_\delta(\boldsymbol{P}^\star)$ also with $\delta = 2$, defined by (C.3). Note that there are $K := \binom{c}{2} - 1 = \frac{c(c-1)}{2} - 1$ neighboring vertices and adjacent edges.

We will now show how perturbing $\boldsymbol{C}$ in just one direction $\boldsymbol{V}$ changes $\boldsymbol{P}^\star$. First, consider a cost matrix that produce $\boldsymbol{P}^\star$, which is defined as $\boldsymbol{C}^\star := \sum_{\boldsymbol{V}_i \in \mathcal{V}_\delta(\boldsymbol{P}^\star)} \boldsymbol{V}_i$, meaning that it is equi-angle from all $\boldsymbol{V} \in \mathcal{V}_\delta(\boldsymbol{P}^\star)$, for $\delta = 2$. By enumerating over all $\boldsymbol{V}$, we may derive $\boldsymbol{C}^\star = \mathbb{1}\mathbb{1}^\top - c\boldsymbol{I}$, along with the fact that $\frac{\langle \boldsymbol{C}^\star, \boldsymbol{V} \rangle}{\|\boldsymbol{C}^\star\|_F \|\boldsymbol{V}\|_F} = \frac{1}{\sqrt{c-1}}$. Define $\boldsymbol{P}_1 = L(\boldsymbol{C}_1)$ with $\boldsymbol{C}_1$ such that $\langle \boldsymbol{V}_1, \boldsymbol{C}_1 \rangle \leq 0$ and $\frac{\langle \boldsymbol{V}_i, \boldsymbol{C}_1 \rangle}{\|\boldsymbol{V}_i\|_F \|\boldsymbol{C}_1\|_F} = \frac{1}{\sqrt{c-1}}$, $\forall i \in 2, \ldots, K$. Define $\boldsymbol{P}_2 = L(\boldsymbol{C}_2)$ with $\boldsymbol{C}_2$ such that $\frac{\langle \boldsymbol{V}_i, \boldsymbol{C}_2 \rangle}{\|\boldsymbol{V}_i\|_F \|\boldsymbol{C}_2\|_F} = \frac{1}{\sqrt{c-1}}$, $\forall i \in [\![K]\!]$. We make the claim that $\boldsymbol{P}_1 = \boldsymbol{P}_2$ and proceed with a proof by contradiction. As mentioned before, $\forall \boldsymbol{V} \in \mathcal{V}_\delta(\boldsymbol{P}^\star)$ for $\boldsymbol{P}^\star = \boldsymbol{I}$, $\delta = 2$ has exactly four non-zero entries, i.e., $V_{ii} = V_{jj} = -1$ and $V_{ij} = V_{ji} = +1$, where $i, j \in [\![c]\!], i \neq j$. Writing out $\langle \boldsymbol{V}, \boldsymbol{C} \rangle$ explicitly, we have $\langle \boldsymbol{V}, \boldsymbol{C} \rangle = -(C_{ii} + C_{jj}) + (C_{ij} + C_{ji})$. To ensure that $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ does not interact with other edges of the polytope, we fix $\langle \boldsymbol{V}_i, \boldsymbol{C} \rangle = \langle \boldsymbol{V}_i, \boldsymbol{C}^\star \rangle = c$ for $i = 2, \ldots, K$. Since we constructed $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ to differ only by the condition $\langle \boldsymbol{V}_1, \boldsymbol{C}_1 \rangle \leq 0$ or $\langle \boldsymbol{V}_1, \boldsymbol{C}_2 \rangle > 0$, and

any $\boldsymbol{V}_i$ affects only four entries of the cost matrix, we may greatly simplify our analysis of $\boldsymbol{C}$ and $\boldsymbol{P}$ to only these four entries. As such, we extract these four entries of $\boldsymbol{P}$, represent it using $\widehat{\boldsymbol{P}} \in \mathcal{B}_2$, and parameterize using $t \in [0, 1]$ it as

$$\widehat{\boldsymbol{P}}(t) = t \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + (1-t) \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{C.5}$$

With this parameterized form, we may reexpress the optimization for $L(\boldsymbol{C})$ as

$$\min_{t \in [0,1]} \; t(C_{ii} + C_{jj}) + (1-t)(C_{ij} + C_{ji}) = \min_{t \in [0,1]} \; \Big(\frac{C_{ii} + C_{jj}}{C_{ij} + C_{ji}} - 1 \Big) t$$

The above minimization has three cases. If $C_{ii} + C_{jj} = C_{ij} + C_{ji}$ then there exists no unique solution t. If $C_{ii} + C_{jj} > C_{ij} + C_{ji}$ then $t = 0$. If $C_{ii} + C_{jj} < C_{ij} + C_{ji}$ then $t = 1$. The first two cases directly corresponds to $\langle \boldsymbol{V}_1, \boldsymbol{C}_1 \rangle \leq 0$, while the third case corresponds to $\langle \boldsymbol{V}_1, \boldsymbol{C}_2 \rangle > 0$. The fact that t is not consistent between all cases demonstrates a contradiction. Moreover, $t = 1$ produces the solution $\widehat{\boldsymbol{P}} = \boldsymbol{I}$, and if this holds for all $\boldsymbol{V} \in \mathcal{V}_\delta(\boldsymbol{P}^\star)$, then all off-diagonal entries must be zero and therefore $\boldsymbol{P}^\star = \boldsymbol{I}$ must be the minimizer for $L(\boldsymbol{C})$. □

A direct consequence of lemma C.1.1 is the following.

Corollary C.1.2. *The solution to the linear program defined by* (C.1) *is* $\boldsymbol{P}^\star = \boldsymbol{I}$ *if the linear cost matrix* $\boldsymbol{C}$ *satisfies the following property*

$$C_{ij} + C_{ji} - C_{ii} - C_{jj} > 0, \;\; \forall i, j \in [\![c]\!], i \neq j. \tag{C.6}$$

Proof. Analyzing (C.4), we observe that any $\boldsymbol{V} \in \mathcal{V}_\delta(\boldsymbol{P}^\star)$ for $\delta = 2$ has only four symmetric non-negative entries, which we condense $\boldsymbol{V}$ and $\boldsymbol{C}$ into $\mathbb{R}^{2\times 2}$ matrices at these four support locations respectively as $\widehat{\boldsymbol{V}} = \begin{bmatrix} -1 & +1 \\ +1 & -1 \end{bmatrix}$ and $\widehat{\boldsymbol{C}} = \begin{bmatrix} C_{ii} & C_{ji} \\ C_{ij} & C_{jj} \end{bmatrix}$. It thus follows

that an explicit computation produces:

$$\langle \boldsymbol{P}, \boldsymbol{C} \rangle \geq 0 \Rightarrow \left\langle \widehat{\boldsymbol{P}}, \widehat{\boldsymbol{C}} \right\rangle = (C_{ij} + C_{ji}) - (C_{ii} + C_{jj}) \geq 0.$$

Since the set $\mathcal{V}_\delta(\boldsymbol{P}^\star)$ spans all permutations between $i, j \in [\![c]\!], i \neq j$, we conclude with (C.6). $\square$

In the variational setting, $C_{ij}(\boldsymbol{R}, \boldsymbol{Q}_{ij})$'s are themselves linearly coupled with $\boldsymbol{R}$ and $\boldsymbol{Q}_{ij}$. The following proposition introduces a trivial criterion on $C_{ij}(\boldsymbol{R}, \boldsymbol{Q}_{ij})$'s to guarantee that $\boldsymbol{P}$ remains unperturbed from $\boldsymbol{P}^\star$. For pedagogical reasons, we shall assume that $n \to \infty$ for this proposition but subsequently relax this.

Proposition C.1.3 (Rotationally invariant disambiguity criterion)**.** *Problem* (6.3) *yields the solution $\boldsymbol{P}^\star$ if, $\forall i, j : i \neq j$, the following criterion is satisfied:*

$$\min_{\boldsymbol{R}, \boldsymbol{Q}_{ij}} C_{ij}(\boldsymbol{R}, \boldsymbol{Q}_{ij}) + \min_{\boldsymbol{R}, \boldsymbol{Q}_{ji}} C_{ji}(\boldsymbol{R}, \boldsymbol{Q}_{ji}) - \min_{\boldsymbol{R}, \boldsymbol{Q}_{ii}} C_{ii}(\boldsymbol{R}, \boldsymbol{Q}_{ii}) - \min_{\boldsymbol{R}, \boldsymbol{Q}_{jj}} C_{jj}(\boldsymbol{R}, \boldsymbol{Q}_{jj}) > 0. \quad \text{(C.7)}$$

Proof. Consider a set of c clusters where clusters i, j satisfy

$$C_{ij}(\boldsymbol{R}'_{ij}, \boldsymbol{Q}'_{ij}) + C_{ji}(\boldsymbol{R}'_{ji}, \boldsymbol{Q}'_{ji}) - C_{ii}(\boldsymbol{R}', \boldsymbol{Q}'_{ii}) - C_{jj}(\boldsymbol{R}', \boldsymbol{Q}'_{jj}) \leq 0, \quad \text{(C.8)}$$

where

$$\begin{aligned}
(\boldsymbol{R}'_{ij}, \boldsymbol{Q}'_{ij}) &:= \operatorname*{arg\,min}_{\boldsymbol{R} \in S(d,d), \boldsymbol{Q}_{ij} \in \mathcal{B}_n} C_{ij}(\boldsymbol{R}), \\
(\boldsymbol{R}'_{ji}, \boldsymbol{Q}'_{ji}) &:= \operatorname*{arg\,min}_{\boldsymbol{R} \in S(d,d), \boldsymbol{Q}_{ji} \in \mathcal{B}_n} C_{ji}(\boldsymbol{R}), \\
(\boldsymbol{R}', \boldsymbol{Q}'_{ii}, \boldsymbol{Q}'_{jj}) &:= \operatorname*{arg\,min}_{\boldsymbol{R} \in S(d,d), \boldsymbol{Q}_{ii}, \boldsymbol{Q}_{jj} \in \mathcal{B}_n} C_{ii}(\boldsymbol{R}, \boldsymbol{Q}_{ii}) + C_{jj}(\boldsymbol{R}, \boldsymbol{Q}_{jj}).
\end{aligned}$$

Since the 2-Wasserstein is a valid metric, its distance between any two clusters must satisfy $C_{ij}(\boldsymbol{R}) \geq 0$ with equality holding if and only if the clusters are exactly similar. If $\boldsymbol{P}^\star =$

$\boldsymbol{I}$, similar clusters are denoted with matching indices, and it must follow that $C_{ii}(\boldsymbol{R}') + C_{jj}(\boldsymbol{R}') = 0$. This implies that (C.8) must be false since $C_{ij}(\boldsymbol{R}'_{ij}), C_{ji}(\boldsymbol{R}'_{ji}) > 0$ for mismatched clusters. Due to this contradiction, the disambiguity criterion (C.7) must hold for all cluster pairs $i, j : i \neq j$. □

This proposition provides a disambiguity criterion, requiring that matched clusters (i.e., C_{ii}, C_{jj}) should be more similar than mismatched clusters (i.e., C_{ij}, C_{ji}) up to some disambiguity *threshold* (in the case of $n \to \infty$, the threshold is 0). To extend this proposition to the finite-sample regime, we require a higher disambiguity-threshold to offset uncertainty due to finite samples.

C.1.2 Part 2: Disambiguity criterion in the finite-sample regime

We utilize a recent p-Wasserstein concentration bound by Weed and Bach [67] that describes finite sample behavior on the Wasserstein distance for data embedded in high-dimensional space, but whose clusters are themselves approximately low-dimensional. We will proceed our analysis with the language of probability measures μ to make our analysis consistent with [67]. We thus define the equivalent measure analogs as follows.

Definition C.1.4. *Let clusters $\boldsymbol{X}_i \in \mathbb{R}^{D \times n_{x,i}}$ and $\boldsymbol{Y}_j \in \mathbb{R}^{D \times n_{y,j}}$ be respectively denoted by empirical measures as*

$$\widehat{\mu}_{x,i} := \frac{1}{n_{x,i}} \sum_{k=1}^{n_{x,i}} \delta_{\boldsymbol{X}_i(k)}, \qquad \widehat{\mu}_{y,i} := \frac{1}{n_{y,j}} \sum_{k=1}^{n_{y,j}} \delta_{\boldsymbol{Y}_j(k)},$$

where $\delta_{\boldsymbol{X}_i(k)}$ refers to a discrete point located at $\boldsymbol{X}_i(k)$. At the limit, we denote the measures as

$$\mu_{x,i} := \lim_{n_{x,i} \to \infty} \frac{1}{n_{x,i}} \sum_{k=1}^{n_{x,i}} \delta_{\boldsymbol{X}_i(k)}, \qquad \mu_{y,j} := \lim_{n_{y,j} \to \infty} \frac{1}{n_{y,j}} \sum_{k=1}^{n_{y,j}} \delta_{\boldsymbol{Y}_j(k)}.$$

Definition C.1.5. *Denote a linear transformation $\boldsymbol{R}$ applied on the measure as*

$$\boldsymbol{R} \circ \widehat{\mu}_{x,i} := \frac{1}{n_{x,i}} \sum_{k=1}^{n_{x,i}} \delta_{\boldsymbol{R}\boldsymbol{X}_i(k)}, \qquad \boldsymbol{R} \circ \mu_{x,i} := \lim_{n_{x,i} \to \infty} \frac{1}{n_{x,i}} \sum_{k=1}^{n_{x,i}} \delta_{\boldsymbol{R}\boldsymbol{X}_i(k)}.$$

The transformed inter-cluster distance between clusters may thus be denoted as

$$\widehat{C}_{ij}(\boldsymbol{R}) := \mathcal{W}_2^2(\boldsymbol{R} \circ \widehat{\mu}_{x,i}, \widehat{\mu}_{y,i}), \qquad \widetilde{C}_{ij}(\boldsymbol{R}) := \mathcal{W}_2^2(\boldsymbol{R} \circ \mu_{x,i}, \mu_{y,i}).$$

Now, we may proceed to state results from [67]. The following result pertains to the sample complexity of measures μ in $\mathbb{R}^D$ that are supported on an approximately low-dimensional set in $\mathbb{R}^d$, where $d \ll D$. First we require some definitions.

Definition C.1.6. *Given a set $S \subseteq \mathbb{R}^d$, let $\mathcal{N}_\varepsilon(S)$ denote the ε-covering number of set S, which is defined as the minimum number m of closed balls $B_1, \ldots, B_m$ of diameter ε such that $S \subseteq \bigcup_{1 \leq i \leq m} B_i$.*

Definition C.1.7. *For any set $S \subseteq \mathbb{R}^d$, the ε-fattening of S is $S_\varepsilon := \{y : D(y, S) \leq \varepsilon\}$.*

Proposition C.1.8 (Weed and Bach [67], Proposition 16)**.** *Let S be a set that satisfies $\mathcal{N}_{\varepsilon'}(S) \leq (3\varepsilon')^{-d}$ for all $\varepsilon' \leq 1/27$ and for some $d > 2p$. Suppose there exists a positive constant σ such that μ satisfies $\mu(S_\varepsilon) \geq 1 - e^{-\varepsilon^2/2\sigma^2}$ for all $\varepsilon > 0$. If $p \log \frac{1}{\sigma} \geq 1/18$, then for all $n \leq (18p\sigma^2 \log \frac{1}{\sigma})^{-d/2}$,*

$$\mathbb{E}[\mathcal{W}_p^p(\mu, \widehat{\mu}_n)] \leq c n^{-p/d},$$

where $c = 27^p(2 + \frac{1}{3^{d/2-p}-1})$.

This proposition states that the degree that μ is concentrated (as parameterized by σ) around set S (approximately supported in low-dimensions) affects how "long" (in terms of n) we can enjoy the fast convergence rate of $n^{-p/d}$. We will leverage this result to obtain the following theorem on cluster correspondence disambiguity with respect to sample

complexity.

Corollary C.1.9. *Let sets $S_{x,i}$ and $S_{y,j}$ satisfy the conditions for S in proposition C.1.8 for some $\sigma_{x,i}, \sigma_{y,j} > 0$ and $d_{x,i}, d_{y,j} > 4$. If $\log \frac{1}{\sigma_{x,i}} \geq \frac{1}{36}$ and $\log \frac{1}{\sigma_{y,j}} \geq \frac{1}{36}$, then for all $n_{x,i} \leq (36\sigma_{x,i}^2 \log \frac{1}{\sigma_{x,i}})^{-d_{x,i}/2}$ and $n_{y,j} \leq (36\sigma_{y,j}^2 \log \frac{1}{\sigma_{y,j}})^{-d_{y,j}/2}$,*

$$\mathbb{E}[|\widetilde{C}_{ij}(\boldsymbol{R}) - \widehat{C}_{ij}(\boldsymbol{R})|] \leq c_{x,i} n_{x,i}^{-2/d_{x,i}} + c_{y,j} n_{y,j}^{-2/d_{y,j}},$$

where

$$c_{z,k} = 729(2 + \frac{1}{3^{d_{z,k}/2-2} - 1}).$$

Proof. Denote μ_i, μ_j as measures and $\widehat{\mu}_i, \widehat{\mu}_j$ as their empirical estimates. By the triangle inequality,

$$\begin{aligned}
\mathbb{E}[\mathcal{W}_2^2(\widehat{\mu}_i, \widehat{\mu}_j)] &\leq \mathbb{E}[\mathcal{W}_2^2(\widehat{\mu}_i, \mu_i) + \mathcal{W}_2^2(\mu_j, \widehat{\mu}_j)] \\
&\leq \mathbb{E}[\mathcal{W}_2^2(\widehat{\mu}_i, \mu_i) + \mathcal{W}_2^2(\mu_i, \mu_j) + \mathcal{W}_2^2(\widehat{\mu}_j, \mu_j)] \\
\Rightarrow \mathbb{E}[|\mathcal{W}_2^2(\mu_i, \mu_j) - \mathcal{W}_2^2(\widehat{\mu}_i, \widehat{\mu}_j)|] &\leq \mathbb{E}[\mathcal{W}_2^2(\widehat{\mu}_i, \mu_i)] + \mathbb{E}[\mathcal{W}_2^2(\widehat{\mu}_j, \mu_j)] \\
\Rightarrow \mathbb{E}[|\widetilde{C}_{ij}(\boldsymbol{R}) - \widehat{C}_{ij}(\boldsymbol{R})|] &\leq c_{x,i} n_{x_i}^{-2/d_{x,i}} + c_{y,j} n_{y,j}^{-2/d_{y,j}},
\end{aligned}$$

where the last line is a direct application of definition C.1.5 and proposition C.1.8. □

Lemma C.1.10. *Let $S_{x,i}, S_{y,i}, S_{x,j}, S_{y,j}$ be sets that satisfy the conditions for S in proposition C.1.8 for some $\sigma_{x,i}, \sigma_{y,i}, \sigma_{x,j}, \sigma_{y,j} > 0$ and $d_{x,i}, d_{y,i}, d_{x,j}, d_{y,j} > 4$. If $\log \frac{1}{\sigma_{z,k}} \geq \frac{1}{36}, \forall z \in \{x, y\}, k \in \{i, j\}$, then for all $n_{z,k} \leq (36\sigma_{z,k}^2 \log \frac{1}{\sigma_{z,k}})^{-d_{z,k}/2}$, $z \in \{x, y\}, k \in \{i, j\}$, cluster correspondences in problem* (6.3) *may be disambiguated to achieve $\boldsymbol{P}^\star = \boldsymbol{I}$ when the following criterion is fulfilled for all $i, j \in [\![c]\!] : i \neq j$:*

$$\mathbb{E}[\min_{\boldsymbol{R}} \widehat{C}_{ij}(\boldsymbol{R}) + \min_{\boldsymbol{R}} \widehat{C}_{ji}(\boldsymbol{R}) - \min_{\boldsymbol{R}} \widehat{C}_{ii}(\boldsymbol{R}) - \min_{\boldsymbol{R}} \widehat{C}_{jj}(\boldsymbol{R})] > 2(B_{x,i} + B_{y,i} + B_{x,j} + B_{y,j}).$$

where the constants are defined as

$$B_{z,k} := c_{z,k} n_{z,k}^{-2/d_{z,k}}, \quad c_{z,k} := 729\Big(2 + \frac{1}{3^{d_{z,k}/2-2} - 1}\Big).$$

Proof. This follows a direct application of criterion (C.7) and corollary C.1.9. □

Proposition C.1.11 (Weed and Bach [67], Proposition 20)**.** *For all* $n \geq 0$ *and* $0 \leq p < \infty$*,*

$$\mathbb{P}[\mathcal{W}_p^p(\mu, \widehat{\mu}_n) \geq \mathbb{E}\mathcal{W}_p^p(\mu, \widehat{\mu}_n) + t] \leq \exp(-2nt^2).$$

Theorem C.1.12. *Define* $\widehat{C}_{ij}^{\star} := \min_{\boldsymbol{R} \in S(D,D)} \widehat{C}_{ij}(\boldsymbol{R})$*. If the conditions in Lemma C.1.10 are satisfied then problem* (6.3) *yields the solution* $\boldsymbol{P}^{\star} = \boldsymbol{I}$ *with probability at least* $1 - \delta$ *if,* $\forall i, j : i \neq j$*, the following criterion is satisfied:*

$$\widehat{C}_{ij}^{\star} + \widehat{C}_{ji}^{\star} - \widehat{C}_{ii}^{\star} - \widehat{C}_{jj}^{\star} \; > \; 2(B_{x,i}(\delta) + B_{y,i}(\delta) + B_{x,j}(\delta) + B_{y,j}(\delta))$$

where

$$B_{z,k}(\delta) := c_{z,k} n_{z,k}^{-\frac{2}{d_{z,k}}} + \sqrt{\log(1/\delta)/2n_{z,k}}, \quad c_{z,k} := 729\Big(2 + \frac{1}{3^{d_{z,k}/2-2} - 1}\Big), \tag{C.9}$$

where $d_{z,k}$ *refers to the intrinsic dimension of the* k*-th cluster from the* z*-th dataset.*

Proof. For some measure μ and its empirical finite-sample estimate $\widehat{\mu}_n$, proposition C.1.11 may be equivalently stated with the choice of $t = \sqrt{\log \frac{1}{\delta}/2n}$ as:

$$|\mathcal{W}_p^p(\mu, \widehat{\mu}_n) - \mathbb{E}\mathcal{W}_p^p(\mu, \widehat{\mu}_n)| \leq \sqrt{\log(1/\delta)/2n},$$

holds with at least probability $1 - \delta$. Under the conditions stated in proposition (C.1.8),

and combining its result with the above relation, we have

$$\mathcal{W}_p^p(\mu, \widehat{\mu}_n) \leq cn^{-p/d} + \sqrt{\log(1/\delta)/2n},$$

where $c = 27^p(2 + \frac{1}{3^{d/2-p}-1})$. Combining this for all terms in the left-hand side of (C.9) yields the stated result. □

C.1.3 Putting everything together

The final proof of Theorem 6.3.1 is a simplified version of Theorem C.1.12's since we assert a stronger (but cleaner) exact low-rank assumption to streamline communication. When the data is *exactly* supported in low-dimensions (as opposed to approximately), the ε-fattening disappears (i.e., $\varepsilon \to 0$) thus any positive $\sigma < \varepsilon$ will send $n \to \infty$, implying that the rapid convergence in dimensions $d \ll D$ holds for $n \to \infty$. Hence an identical result holds, with the sole condition that $d > 4$. □

C.2 Worst-case alignability: geometric conditions

We apply a very recent perturbation bound for the Procrustes problem developed by Arias-Castro et al. [213] to subsequently state a cluster-based alignment bound. First, we outline the perturbation bound for the classical Procrustes problem below.

Theorem C.2.1 (Procrustes perturbation bounds, Theorem 1 [213])**.** *Consider short matrices $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{d\times n}$ with $d < n$ and $\boldsymbol{X}$ having full rank. Set $\varepsilon^2 = \left\|\boldsymbol{Y}^\top\boldsymbol{Y} - \boldsymbol{X}^\top\boldsymbol{X}\right\|_p$, where $\|\cdot\|_p$ denotes the Schatten p-norm. Denote the singular value decomposition of $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\boldsymbol{\Sigma}$ contains diagonal elements $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_d > 0 = \cdots = 0$, and let $\boldsymbol{X}^\dagger$ be the pseudo-inverse of $\boldsymbol{X}$, i.e., $\boldsymbol{X}^\dagger = \boldsymbol{U}\boldsymbol{\Sigma}^\dagger\boldsymbol{V}^\top$, where $\boldsymbol{\Sigma}^\dagger = \operatorname{diag}(\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_d^{-1}, 0, \ldots, 0)$. If $\left\|\boldsymbol{X}^\dagger\right\| \varepsilon \leq \frac{1}{\sqrt{2}}(\|\boldsymbol{X}\| \left\|\boldsymbol{X}^\dagger\right\|)^{-1/2}$ then*

$$\min_{\boldsymbol{R}\in S(d,d)} \|\boldsymbol{R}\boldsymbol{X} - \boldsymbol{Y}\|_p \leq (\|\boldsymbol{X}\| \left\|\boldsymbol{X}^\dagger\right\| + 2) \left\|\boldsymbol{X}^\dagger\right\| \varepsilon^2.$$

Directly applying Theorem C.2.1 using the Schatten 2-norm (i.e., the Frobenius norm) yields

$$\begin{aligned}\min_{\boldsymbol{R}\in S(d,d)} \|\boldsymbol{R}\boldsymbol{X}-\boldsymbol{Y}\|_F^2 &= \min_{\boldsymbol{R}\in S(d,d)} \sum_{i=1}^{c} \operatorname{tr}(\boldsymbol{Q}_{ii}^\top \boldsymbol{X}_i^\top \boldsymbol{X}_i \boldsymbol{Q}_{ii} + \boldsymbol{Y}_i^\top \boldsymbol{Y}_i - 2\boldsymbol{Y}_i \boldsymbol{Q}_{ii}^\top \boldsymbol{X}_i^\top \boldsymbol{R}^\top) \\ &\leq ((\|\boldsymbol{X}\| \left\|\boldsymbol{X}^\dagger\right\| + 2) \left\|\boldsymbol{X}^\dagger\right\| \varepsilon^2)^2 := B^2,\end{aligned}$$

$$\Rightarrow \max_{\boldsymbol{R}\in S(d,d)} \sum_{i=1}^{c} \operatorname{tr}(2\boldsymbol{Y}_i \boldsymbol{Q}_{ii}^\top \boldsymbol{X}_i^\top \boldsymbol{R}^\top) \geq \operatorname{tr}(\boldsymbol{X}\boldsymbol{X}^\top + \boldsymbol{Y}\boldsymbol{Y}^\top) - B^2,$$

where $\varepsilon^2 := \left\|\boldsymbol{Y}^\top\boldsymbol{Y} - \boldsymbol{X}^\top\boldsymbol{X}\right\|_F$ and under the conditions that $\boldsymbol{X}$ is full rank and $\left\|\boldsymbol{X}^\dagger\right\| \varepsilon \leq \frac{1}{\sqrt{2}}(\|\boldsymbol{X}\| \left\|\boldsymbol{X})^\dagger\right\|)^{-1/2}$. When the criterion given by corollary C.1.3 is satisfied, we are guaranteed cluster-correspondences $\boldsymbol{P}^\star = \boldsymbol{I}$. Therefore $\sum_{ij} P_{ij}C_{ij}(\boldsymbol{R}) = \sum_i C_{ii}(\boldsymbol{R})$. We utilize this to lower bound $\sum_i C_{ii}(\boldsymbol{R})$ as follows:

$$\begin{aligned}&\min_{\boldsymbol{R}\in S(d,d)} \sum_{i=1}^{c} C_{ii}(\boldsymbol{R}) \\ =&\min_{\boldsymbol{R}\in S(d,d)} \sum_{i=1}^{c} \operatorname{tr}(\boldsymbol{X}_i \operatorname{diag}(\boldsymbol{Q}_{ii}\mathbb{1})\boldsymbol{X}_i^\top + \boldsymbol{Y}_i \operatorname{diag}(\boldsymbol{Q}_{ii}^\top\mathbb{1})\boldsymbol{Y}_j^\top - 2\boldsymbol{Y}_i\boldsymbol{Q}_{ii}^\top\boldsymbol{X}_i^\top\boldsymbol{R}^\top) \\ =&\sum_{i=1}^{c} \operatorname{tr}(\boldsymbol{X}_i \operatorname{diag}(\boldsymbol{Q}_{ii}\mathbb{1})\boldsymbol{X}_i^\top + \boldsymbol{Y}_i \operatorname{diag}(\boldsymbol{Q}_{ii}^\top\mathbb{1})\boldsymbol{Y}_j^\top) - \max_{\boldsymbol{R}\in S(d,d)} \sum_{i=1}^{c} \operatorname{tr}(2\boldsymbol{Y}_j\boldsymbol{Q}_{ii}^\top\boldsymbol{X}_i^\top\boldsymbol{R}^\top) \\ \leq&B^2 + \sum_{i=1}^{c} \operatorname{tr}(\boldsymbol{X}_i(\operatorname{diag}(\boldsymbol{Q}_{ii}\mathbb{1}) - \boldsymbol{Q}_{ii}\boldsymbol{Q}_{ii}^\top)\boldsymbol{X}_i^\top + \boldsymbol{Y}_j(\operatorname{diag}(\boldsymbol{Q}_{ii}\mathbb{1}) - \boldsymbol{I})\boldsymbol{Y}_j^\top) \\ =&B^2 + \sum_{i=1}^{c} \operatorname{tr}(\boldsymbol{X}_i(1/n - \boldsymbol{Q}_{ii}\boldsymbol{Q}_{ii}^\top)\boldsymbol{X}_i^\top + \boldsymbol{Y}_j(1/n - \boldsymbol{I})\boldsymbol{Y}_j^\top).\end{aligned}$$

□

C.3 Alignment error bounds of 2-Wasserstein under Stiefel constraints

To ease notation, let $\boldsymbol{A} = \boldsymbol{Y}_j\boldsymbol{Q}_{ij}^\top\boldsymbol{X}_i^\top$, and $\boldsymbol{C} = \frac{1}{n}\boldsymbol{X}_i\boldsymbol{X}_i^\top + \frac{1}{n}\boldsymbol{Y}_j\boldsymbol{Y}_j^\top$. Let $\boldsymbol{A}$ be decomposed by the singular-value decomposition as $\boldsymbol{A} = \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Sigma}}\widetilde{\boldsymbol{V}}^\top$. Lastly, let $\boldsymbol{U} = [\boldsymbol{U}', \boldsymbol{U}'']$ and

$\boldsymbol{V} = [\boldsymbol{V}', \boldsymbol{V}'']$, where $\langle \boldsymbol{U}'', \boldsymbol{U}' \rangle = \langle \boldsymbol{V}'', \boldsymbol{V}' \rangle = 0$. Then it follows that

$$\operatorname{tr}(\boldsymbol{U}^\top \widetilde{\boldsymbol{U}} \widetilde{\boldsymbol{\Sigma}} \widetilde{\boldsymbol{V}}^\top \boldsymbol{V}) \leq \operatorname{tr}(\widetilde{\boldsymbol{\Sigma}}) = \operatorname{tr}(\widetilde{\boldsymbol{U}}^\top \widetilde{\boldsymbol{U}} \widetilde{\boldsymbol{\Sigma}} \widetilde{\boldsymbol{V}}^\top \widetilde{\boldsymbol{V}})$$

(C.10)

$$\begin{aligned}
\operatorname{tr}(\boldsymbol{U}^\top \boldsymbol{A} \boldsymbol{V}) &\leq \operatorname{tr}(\widetilde{\boldsymbol{U}}^\top \boldsymbol{A} \widetilde{\boldsymbol{V}}) \\
\operatorname{tr}\left(\boldsymbol{C} - 2\boldsymbol{U}'^\top \boldsymbol{A} \boldsymbol{V}' - 2\boldsymbol{U}''^\top \boldsymbol{A} \boldsymbol{V}''\right) &\geq \operatorname{tr}\left(\boldsymbol{C} - 2\widetilde{\boldsymbol{U}}^\top \boldsymbol{A} \widetilde{\boldsymbol{V}}\right) \\
\min_{\substack{\boldsymbol{U}'', \boldsymbol{V}'' \in S(d, d-r): \\ \langle \boldsymbol{U}', \boldsymbol{U}'' \rangle = \langle \boldsymbol{V}', \boldsymbol{V}'' \rangle = 0}} \operatorname{tr}\left(\boldsymbol{C} - 2\boldsymbol{A}\boldsymbol{V}'\boldsymbol{U}'^\top - 2\boldsymbol{A}\boldsymbol{V}''\boldsymbol{U}''^\top\right) &\geq \min_{\widetilde{\boldsymbol{U}}, \widetilde{\boldsymbol{V}} \in S(d,d)} \operatorname{tr}\left(\boldsymbol{C} - 2\boldsymbol{A}\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{U}}^\top\right) \\
\min_{\boldsymbol{R} \in \mathcal{T}(\boldsymbol{U}', \boldsymbol{V}')} C_{ij}(\boldsymbol{R}) &\geq \min_{\boldsymbol{R} \in S(d,d)} C_{ij}(\boldsymbol{R}).
\end{aligned}$$

What remains is for us to show the condition for equality. From (C.10), we have that

$$\operatorname{tr}(\boldsymbol{U}^\top \widetilde{\boldsymbol{U}} \widetilde{\boldsymbol{\Sigma}} \widetilde{\boldsymbol{V}}^\top \boldsymbol{V}) = \operatorname{tr}(\widetilde{\boldsymbol{V}}^\top \boldsymbol{V} \boldsymbol{U}^\top \widetilde{\boldsymbol{U}} \widetilde{\boldsymbol{\Sigma}}) \leq \operatorname{tr}(\widetilde{\boldsymbol{\Sigma}}),$$

with equality holding if $\widetilde{\boldsymbol{V}}^\top \boldsymbol{V} \boldsymbol{U}^\top \widetilde{\boldsymbol{U}} = \boldsymbol{I}$, implying that $\widetilde{\boldsymbol{U}}^\top \boldsymbol{U} = \widetilde{\boldsymbol{V}}^\top \boldsymbol{V}$, which imply that

$$\langle \widetilde{\boldsymbol{U}}, \boldsymbol{U}' \rangle = \langle \widetilde{\boldsymbol{V}}, \boldsymbol{V}' \rangle, \qquad \langle \widetilde{\boldsymbol{U}}, \boldsymbol{U}'' \rangle = \langle \widetilde{\boldsymbol{V}}, \boldsymbol{V}'' \rangle,$$

which are obtained via the substitutions $\boldsymbol{U} = [\boldsymbol{U}', \boldsymbol{U}'']$ and $\boldsymbol{V} = [\boldsymbol{V}', \boldsymbol{V}'']$. $\square$

REFERENCES

[1] S. Chen and D. Donoho, "Basis pursuit," Tech. Rep., 1994.

[2] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, p. 607, 1996.

[3] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[4] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[5] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 83–91, 2008.

[6] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[7] H. W. Sorenson, "Least-squares estimation: From gauss to kalman," *IEEE spectrum*, vol. 7, no. 7, pp. 63–68, 1970.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[9] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," *ArXiv preprint arXiv:1610.05256*, 2016.

[10] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[11] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM journal on computing*, vol. 24, no. 2, pp. 227–234, 1995.

[12] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[13] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 877–905, 2008.

[14] P. Garrigues and B. A. Olshausen, "Group sparse coding with a Laplacian scale mixture prior," in *Advances in Neural Information Processing Systems*, 2010, pp. 676–684.

[15] A. S. Charles, A. Balavoine, and C. J. Rozell, "Dynamic filtering of time-varying sparse signals via ℓ_1 minimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 21, pp. 5644–5656, 2015.

[16] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[17] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010.

[18] L. Condat, "Discrete total variation: New definition and minimization," *SIAM Journal on Imaging Sciences*, vol. 10, no. 3, pp. 1258–1290, 2017.

[19] L. A. Shepp and B. F. Logan, "The Fourier reconstruction of a head section," *IEEE Transactions on nuclear science*, vol. 21, no. 3, pp. 21–43, 1974.

[20] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[21] H. H. Bauschke, P. L. Combettes, *et al.*, *Convex analysis and monotone operator theory in hilbert spaces*. Springer, 2011, vol. 408.

[22] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[23] P.-L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM Journal on Numerical Analysis*, vol. 16, no. 6, pp. 964–979, 1979.

[24] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of mathematical imaging and vision*, vol. 40, no. 1, pp. 120–145, 2011.

[25] J. Eckstein, "Splitting methods for monotone operators with applications to parallel optimization," PhD thesis, Massachusetts Institute of Technology, 1989.

[26] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM journal on control and optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[27] N. Parikh, S. Boyd, *et al.*, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[28] J. Eckstein and D. P. Bertsekas, "On the Douglasâăť Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1, pp. 293–318, 1992.

[29] J. Douglas and H. H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," *Transactions of the American mathematical Society*, vol. 82, no. 2, pp. 421–439, 1956.

[30] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 337–364, 2016.

[31] M. S. Almeida and M. Figueiredo, "Deconvolving images with unknown boundaries using the alternating direction method of multipliers," *IEEE Transactions on Image processing*, vol. 22, no. 8, pp. 3074–3086, 2013.

[32] Y. Wang, W. Yin, and J. Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," *Journal of Scientific Computing*, vol. 78, no. 1, pp. 29–63, 2019.

[33] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR computational mathematics and mathematical physics*, vol. 7, no. 3, pp. 200–217, 1967.

[34] A. S. Nemirovsky and D. B. Yudin, "Problem complexity and method efficiency in optimization.," *Optimization*, 1983.

[35] A. Beck and M. Teboulle, "Mirror descent and nonlinear projected subgradient methods for convex optimization," *Operations Research Letters*, vol. 31, no. 3, pp. 167–175, 2003.

[36] H. Wang and A. Banerjee, "Bregman alternating direction method of multipliers," in *Advances in Neural Information Processing Systems*, 2014, pp. 2816–2824.

[37] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International journal of computer vision*, vol. 40, no. 2, pp. 99–121, 2000.

[38] G. Monge, *Mémoire sur la théorie des déblais et des remblais*. Paris: De l'Imprimerie Royale, 1781.

[39] L. V. Kantorovich, "On the translocation of masses," in *Dokl. Akad. Nauk SSSR*, vol. 37, 1942, pp. 199–201.

[40] C. Villani, *Optimal transport: Old and new*. Springer Science & Business Media, 2008, vol. 338.

[41] F. Santambrogio, "Optimal transport for applied mathematicians," *Birkäuser, NY*, vol. 55, pp. 58–63, 2015.

[42] L. Ambrosio, N. Gigli, and G. Savaré, *Gradient flows: In metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

[43] G. Peyré, M. Cuturi, *et al.*, "Computational optimal transport," *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[44] J. Lellmann, D. A. Lorenz, C. SchoÌĹnlieb, and T. Valkonen, "Imaging with kantorovich–rubinstein discrepancy," *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 2833–2859, 2014.

[45] I. Abraham, R. Abraham, M. Bergounioux, and G. Carlier, "Tomographic reconstruction from a few views: A multi-marginal optimal transport approach," *Applied Mathematics & Optimization*, vol. 75, no. 1, pp. 55–73, 2017.

[46] J. Karlsson and A. Ringh, "Generalized sinkhorn iterations for regularizing inverse problems using optimal mass transport," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1935–1962, 2017.

[47] P. Indyk and E. Price, "K-median clustering, model-based compressive sensing, and sparse recovery for earth mover distance," in *Proceedings of the forty-third annual ACM symposium on Theory of computing*, ACM, 2011, pp. 627–636.

[48] R. Sandler and M. Lindenbaum, "Nonnegative matrix factorization with earth mover's distance metric for image analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1590–1602, 2011.

[49] A. Rolet, M. Cuturi, and G. Peyré, "Fast dictionary learning with a smoothed wasserstein loss," in *Artificial Intelligence and Statistics*, 2016, pp. 630–638.

[50] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," in *Advances in Neural Information Processing Systems*, 2017, pp. 3730–3739.

[51] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1853–1865, 2017.

[52] S. Ferradans, N. Papadakis, G. Peyré, and J.-F. Aujol, "Regularized discrete optimal transport," *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1853–1882, 2014.

[53] W. Li, E. K. Ryu, S. Osher, W. Yin, and W. Gangbo, "A parallel method for earth moverâăźs distance," *Journal of Scientific Computing*, vol. 75, no. 1, pp. 182–197, 2018.

[54] M. Beckmann, "A continuous model of transportation," *Econometrica: Journal of the Econometric Society*, pp. 643–660, 1952.

[55] A. Figalli, "The optimal partial transport problem," *Archive for rational mechanics and analysis*, vol. 195, no. 2, pp. 533–560, 2010.

[56] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard, "Scaling algorithms for unbalanced transport problems," *ArXiv preprint arXiv:1607.05816*, 2016.

[57] M. Liero, A. Mielke, and G. Savaré, "Optimal entropy-transport problems and a new hellinger–kantorovich distance between positive measures," *Inventiones mathematicae*, vol. 211, no. 3, pp. 969–1117, 2018.

[58] J.-D. Benamou, "Numerical resolution of an âăijunbalancedâăi mass transport problem," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 37, no. 5, pp. 851–868, 2003.

[59] L. Caffarelli and R. J. McCann, "Free boundaries in optimal transport and monge-ampere obstacle problems," *Annals of mathematics*, vol. 171, no. 2, pp. 673–730, 2010.

[60] B. Piccoli and F. Rossi, "On properties of the generalized wasserstein distance," *Archive for Rational Mechanics and Analysis*, vol. 222, no. 3, pp. 1339–1365, 2016.

[61] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard, "Unbalanced optimal transport: Geometry and kantorovich formulation," *ArXiv preprint arXiv:1508.05216*, 2015.

[62] L. Gurobi Optimization, *Gurobi optimizer reference manual*, 2018.

[63] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in neural information processing systems*, 2013, pp. 2292–2300.

[64] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific Journal of Mathematics*, vol. 21, no. 2, pp. 343–348, 1967.

[65] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, "Iterative bregman projections for regularized transportation problems," *SIAM Journal on Scientific Computing*, vol. 37, no. 2, A1111–A1138, 2015.

[66] R. Dudley, "The speed of mean glivenko-cantelli convergence," *The Annals of Mathematical Statistics*, vol. 40, no. 1, pp. 40–50, 1969.

[67] J. Weed and F. Bach, "Sharp asymptotic and finite-sample rates of convergence of empirical measures in wasserstein distance," *ArXiv preprint arXiv:1707.00087*, 2017.

[68] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Advances in neural information processing systems*, 2007, pp. 513–520.

[69] A. Genevay, L. Chizat, F. Bach, M. Cuturi, and G. Peyré, "Sample complexity of sinkhorn divergences," *ArXiv preprint arXiv:1810.02733*, 2018.

[70] A. E. Vickers and R. L. Fisher, "Organ slices for the evaluation of human drug toxicity," *Chemico-Biological Interactions*, vol. 150, no. 1, pp. 87–96, 2004.

[71] H Kamioka, T Honjo, and T Takano-Yamamoto, "A three-dimensional distribution of osteocyte processes revealed by the combination of confocal laser scanning microscopy and differential interference contrast microscopy," *Bone*, vol. 28, no. 2, pp. 145–149, 2001.

[72] S. J. Mulligan and B. A. MacVicar, "Calcium transients in astrocyte endfeet cause cerebrovascular constrictions," *Nature*, vol. 431, no. 7005, pp. 195–199, 2004.

[73] A. Bussek, E. Wettwer, T. Christ, H. Lohmann, P. Camelliti, and U. Ravens, "Tissue slices from adult mammalian hearts as a model for pharmacological drug testing," *Cellular Physiology and Biochemistry*, vol. 24, no. 5-6, pp. 527–536, 2009.

[74] I. Simon, C. R. Pound, A. W. Partin, J. Q. Clemens, and W. A. Christens-Barry, "Automated image analysis system for detecting boundaries of live prostate cancer cells," *Cytometry*, vol. 31, no. 4, pp. 287–294, 1998.

[75] K. Wu, D. Gauthier, and M. D. Levine, "Live cell image segmentation," *IEEE Transactions on biomedical engineering*, vol. 42, no. 1, pp. 1–12, 1995.

[76] A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillén-Aghion, J.-C. Olivo-Marin, and C. Zimmer, "Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces," *IEEE Transactions on Image Processing*, vol. 14, no. 9, pp. 1396–1410, 2005.

[77] F. Yang, M. A. Mackey, F. Ianzini, G. Gallardo, and M. Sonka, "Cell segmentation, tracking, and mitosis detection using temporal context," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2005, pp. 302–309.

[78] D. Padfield, J. Rittscher, N. Thomas, and B. Roysam, "Spatio-temporal cell cycle phase analysis using level sets and fast marching methods," *Medical Image Analysis*, vol. 13, no. 1, pp. 143–155, 2009.

[79] O. Dzyubachyk, W. Niessen, and E. Meijering, "A variational model for level-set based cell tracking in time-lapse fluorescence microscopy images," in *4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007.*, IEEE, 2007, pp. 97–100.

[80] C. Zimmer, E. Labruyere, V. Meas-Yedid, N. Guillen, and J.-C. Olivo-Marin, "Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: A tool for cell-based drug testing," *IEEE transactions on medical imaging*, vol. 21, no. 10, pp. 1212–1221, 2002.

[81] M. Maska, O. Danek, S. Garasa, A. Rouzaut, A. Muñoz-Barrutia, and C. Ortiz-de Solorzano, "Segmentation and shape tracking of whole fluorescent cells based on the chan–vese model," *IEEE transactions on medical imaging*, vol. 32, no. 6, pp. 995–1006, 2013.

[82] M. Maska, V. Ulman, D. Svoboda, P. Matula, P. Matula, C. Ederra, A. Urbiola, T. España, S. Venkatesan, D. M. Balak, *et al.*, "A benchmark for comparison of cell tracking algorithms," *Bioinformatics*, vol. 30, no. 11, pp. 1609–1617, 2014.

[83] V. Ulman, M. Maska, K. E. Magnusson, O. Ronneberger, C. Haubold, N. Harder, P. Matula, P. Matula, D. Svoboda, M. Radojevic, *et al.*, "An objective comparison of cell-tracking algorithms," *Nature Methods*, 2017.

[84] D. Padfield, J. Rittscher, and B. Roysam, "Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis," *Medical Image Analysis*, vol. 15, no. 4, pp. 650–668, 2011.

[85] O. P. Hamill, A Marty, E. Neher, B. Sakmann, and F. Sigworth, "Improved patch-clamp techniques for high-resolution current recording from cells and cell-free membrane patches," *Pflügers Archiv European journal of physiology*, vol. 391, no. 2, pp. 85–100, 1981.

[86] G. Stuart, H. Dodt, and B. Sakmann, "Patch-clamp recordings from the soma and dendrites of neurons in brain slices using infrared video microscopy," *Pflügers Archiv European Journal of Physiology*, vol. 423, no. 5, pp. 511–518, 1993.

[87] S. B. Kodandaramaiah, G. T. Franzesi, B. Y. Chow, E. S. Boyden, and C. R. Forest, "Automated whole-cell patch-clamp electrophysiology of neurons in vivo," *Nature methods*, vol. 9, no. 6, p. 585, 2012.

[88] I Kolb, W. Stoy, E. Rousseau, O. Moody, A Jenkins, and C. Forest, "Cleaning patch-clamp pipettes for immediate reuse," *Scientific Reports*, vol. 6, 2016.

[89] J. Lee and C. J. Rozell, "Precision cell boundary tracking on dic microscopy video for patch clamping," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, IEEE, 2017, pp. 1048–1052.

[90] J. Lee, I. Kolb, C. Forest, and C. Rozell, "Cell membrane tracking in live brain tissue with differential interference contrast (DIC) microscopy," in *Society for Neuroscience Annual Meeting*, Washington, D.C., Nov. 2017.

[91] J. Lee, I. Kolb, C. R. Forest, and C. J. Rozell, "Cell membrane tracking in living brain tissue using differential interference contrast microscopy," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1847–1861, 2018.

[92] J. Lee and C. J. Rozell, "Fast admm solver for reweighted total variation image deconvolution and inpainting," in *Signal and Information Processing (GlobalSIP), 2017 IEEE Global Conference on*, IEEE, 2017, pp. 1145–1149.

[93] J. Lee and C. J. Rozell, *Systems and methods for cell membrane identification and tracking, and technique automation using the same*, US Patent App. 16/116,192, Feb. 2019.

[94] I. Kolb, J. Lee, A Felouzis, C. Landry, M. Yip, C. Lewallen, W. Stoy, C. Rozell, and C. Forest, "The patcherBot: A walk-away automated patch-clamp electrophysiology system," in *Society for Neuroscience Annual Meeting*, Washington, D.C., Nov. 2017.

[95] I. Kolb, C. R. Landry, M. C. Yip, C. F. Lewallen, W. A. Stoy, J. Lee, A. Felouzis, B. Yang, E. S. Boyden, C. J. Rozell, *et al.*, "Patcherbot: A single-cell electrophysiology robot for adherent cells and brain slices," *Journal of neural engineering*, vol. 16, no. 4, p. 046 003, 2019.

[96] K. Li and T. Kanade, "Nonnegative mixed-norm preconditioning for microscopy image segmentation," in *Information Processing in Medical Imaging*, Springer, 2009, pp. 362–373.

[97] A. Charles, J. Lee, N. Bertrand, and C. Rozell, "Dynamic filtering with Earth Mover's Distance regularization," in *Signal Processing with Adaptive Sparse Structured Representations (SPARS) Workshop*, Lisbon, Portugal, Jun. 2017.

[98] U. S.G.S. A. Federal Standard, *Telecommunications: Glossary of telecommunication terms*, 1996.

[99] E. Van Munster, L. Van Vliet, and J. Aten, "Reconstruction of optical pathlength distributions from images obtained by a wide-field differential interference contrast microscope," *Journal of Microscopy*, vol. 188, no. 2, pp. 149–157, 1997.

[100] N. N. Kachouie, P. Fieguth, and E. Jervis, "Watershed deconvolution for cell segmentation," in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, 2008, pp. 375–378.

[101] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2015, pp. 234–241.

[102] Z. Yin, R. Bise, M. Chen, and T. Kanade, "Cell segmentation in microscopy imagery using a bag of local Bayesian classifiers," in *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, IEEE, 2010, pp. 125–128.

[103] A. Kuijper and B. Heise, "An automatic cell segmentation method for differential interference contrast microscopy," in *19th International Conference on Pattern Recognition, 2008. ICPR 2008*, IEEE, 2008, pp. 1–4.

[104] B. Obara, M. A. Roberts, J. P. Armitage, and V. Grau, "Bacterial cell identification in differential interference contrast microscopy images," *BMC Bioinformatics*, vol. 14, no. 1, p. 134, 2013.

[105] B. Heise, A. Sonnleitner, and E. P. Klement, "DIC image reconstruction on large cell scans," *Microscopy Research and Technique*, vol. 66, no. 6, pp. 312–320, 2005.

[106] M. Arnison, C. Cogswell, N. Smith, P. Fekete, and K. Larkin, "Using the Hilbert transform for 3D visualization of differential interference contrast microscope images," *Journal of Microscopy*, vol. 199, no. 1, pp. 79–84, 2000.

[107] Z. Yin and T. Kanade, "Restoring artifact-free microscopy image sequences," in *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, IEEE, 2011, pp. 909–913.

[108] A. Lehmussola, P. Ruusuvuori, J. Selinummi, T. Rajala, and O. Yli-Harja, "Synthetic images of high-throughput microscopy for validation of image analysis methods," *Proceedings of the IEEE*, vol. 96, no. 8, pp. 1348–1360, 2008.

[109] D. Svoboda and V. Ulman, "Generation of synthetic image datasets for time-lapse fluorescence microscopy," in *Image Analysis and Recognition*, Springer, 2012, pp. 473–482.

[110] S. Rajaram, B. Pavie, N. E. Hac, S. J. Altschuler, and L. F. Wu, "SimuCell: A flexible framework for creating synthetic microscopy images," *Nature Methods*, vol. 9, no. 7, pp. 634–635, 2012.

[111] R. F. Murphy, "Cellorganizer: Image-derived models of subcellular organization and protein distribution," *Methods in Cell Biology*, vol. 110, p. 179, 2012.

[112] L. Martins, J. Fonseca, and A. Ribeiro, "'miSimBa'âăt'a simulator of synthetic time-lapsed microscopy images of bacterial cells," in *2015 IEEE 4th Portuguese Meeting on Bioengineering (ENBENG)*, IEEE, 2015, pp. 1–6.

[113] Q. Wu, I. Kolb, B. M. Callahan, Z. Su, W. Stoy, S. B. Kodandaramaiah, R. L. Neve, H. Zeng, E. S. Boyden, C. R. Forest, *et al.*, "Integration of autopatching with automated pipette and cell detection in vitro," *Journal of Neurophysiology*, jn–00 386, 2016.

[114] Z Pincus and J. Theriot, "Comparison of quantitative methods for cell-shape analysis," *Journal of Microscopy*, vol. 227, no. 2, pp. 140–156, 2007.

[115] A. Lehmussola, P. Ruusuvuori, J. Selinummi, H. Huttunen, and O. Yli-Harja, "Computational framework for simulating fluorescence microscope images with cell populations," *IEEE transactions on medical imaging*, vol. 26, no. 7, pp. 1010–1016, 2007.

[116] D. Svoboda, M. Kozubek, and S. Stejskal, "Generation of digital phantoms of cell nuclei and simulation of image formation in 3d image cytometry," *Cytometry Part A*, vol. 75, no. 6, pp. 494–509, 2009.

[117] K. Perlin, "An image synthesizer," *ACM Siggraph Computer Graphics*, vol. 19, no. 3, pp. 287–296, 1985.

[118] A. Lehmussola, J. Selinummi, P. Ruusuvuori, A. Niemisto, and O. Yli-Harja, "Simulating fluorescent microscope images of cell populations," in *27th Annual International Conference of the Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005.*, IEEE, 2006, pp. 3153–3156.

[119] Mathworks. (2016). Creating a gallery of transformed images, image 9: Apply barrel transformation to checkerboard, (visited on 06/28/2016).

[120] J. C. Russ and F. B. Neal, *The image processing handbook*, 7th. Boca Raton, FL, USA: CRC Press, Inc., 2015, ISBN: 149874026X, 9781498740265.

[121] E. Olson, "Particle shape factors and their use in image analysis part 1: Theory," *Journal of GXP Compliance*, vol. 15, no. 3, pp. 85–96, 2011.

[122] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[123] X. Lan, A. J. Ma, and P. C. Yuen, "Multi-cue visual tracking using robust feature-level fusion based on joint sparse representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1194–1201.

[124] X. Lan, A. J. Ma, P. C. Yuen, and R. Chellappa, "Joint sparse representation and robust feature-level fusion for multi-cue visual tracking," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5826–5841, 2015.

[125] X. Lan, S. Zhang, and P. C. Yuen, "Robust joint discriminative feature learning for visual tracking.," in *IJCAI*, 2016, pp. 3403–3410.

[126] X. Lan, P. C. Yuen, and R. Chellappa, "Robust mil-based feature template learning for object tracking.," in *AAAI*, 2017, pp. 4118–4125.

[127] A. S. Charles and C. J. Rozell, "Spectral superresolution of hyperspectral imagery using reweighted ℓ_1 spatial filtering," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 3, pp. 602–606, 2013.

[128] M. Grant and S. Boyd, *CVX: Matlab software for disciplined convex programming, version 2.1*, `http://cvxr.com/cvx`, Mar. 2014.

[129] M. C. Grant and S. P. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds., `http://stanford.edu/~boyd/graph_dcp.html`, Springer-Verlag Limited, 2008, pp. 95–110.

[130] S. R. Becker, E. J. Candès, and M. C. Grant, "Templates for convex cone problems with applications to sparse signal recovery," *Mathematical Programming Computation*, vol. 3, no. 3, pp. 165–218, 2011.

[131] J. Shen and T. F. Chan, "Mathematical models for local nontexture inpaintings," *SIAM Journal on Applied Mathematics*, vol. 62, no. 3, pp. 1019–1043, 2002.

[132] M. Tao, J. Yang, and B. He, "Alternating direction algorithms for total variation deconvolution in image reconstruction," *TR0918, Department of Mathematics, Nanjing University*, 2009.

[133] J.-F. Cai, S. Osher, and Z. Shen, "Split Bregman methods and frame based image restoration," *Multiscale modeling & simulation*, vol. 8, no. 2, pp. 337–369, 2009.

[134] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2345–2356, 2010.

[135] R. H. Chan, M. Tao, and X. Yuan, "Constrained total variation deblurring models and fast algorithms based on alternating direction method of multipliers," *SIAM Journal on imaging Sciences*, vol. 6, no. 1, pp. 680–697, 2013.

[136] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *Mathematical Programming*, pp. 1–35,

[137] N. Corporation, *CUDA 8 performance overview*, Nov. 2016.

[138] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, "An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 681–695, 2011.

[139] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

[140] R. Jordan, D. Kinderlehrer, and F. Otto, "The variational formulation of the fokker–planck equation," *SIAM journal on mathematical analysis*, vol. 29, no. 1, pp. 1–17, 1998.

[141] G. Peyré, "Entropic approximation of wasserstein gradient flows," *SIAM Journal on Imaging Sciences*, vol. 8, no. 4, pp. 2323–2351, 2015.

[142] A. Charles, N. Bertrand, J. Lee, and C. Rozell, "Earth-Mover's Distance as a tracking regularizer," in *IEEE INTERNATIONAL WORKSHOP ON COMPUTATIONAL ADVANCES IN MULTI-SENSOR ADAPTIVE PROCESSING (CAMSAP)*, Curaçao, Dutch Antilles, Dec. 2017.

[143] N. Bertrand, A. Charles, J. Lee, P. Dunn, and C. Rozell, "Earth Mover's Distance as a dynamics regularizer for sparse signal tracking," Jun. 2018, Submitted.

[144] E. K. Ryu, W. Li, P. Yin, and S. Osher, "Unbalanced and partial L_1 monge–kantorovich problem: A scalable parallel first-order method," *Journal of Scientific Computing*, vol. 75, no. 3, pp. 1596–1613, 2018.

[145] J. Lee, A. Charles, N. Bertrand, and C. Rozell, "An optimal transport tracking regularizer," in *Neural Information Processing Systems (NIPS) Workshop, Optimal Transport and Machine Learning*, Long Beach, CA, Dec. 2017.

[146] N. Bertrand, J. Lee, A. Charles, P. Dunn, and C. Rozell, "Sparse dynamic filtering via earth mover's distance regularization," in *PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING (ICASSP)*, Calgary, Alberta, Canada, Apr. 2018.

[147] M. S. Asif and J. Romberg, "Dynamic updating for ℓ_1 minimization," *IEEE Journal of selected topics in signal processing*, vol. 4, no. 2, pp. 421–434, 2010.

[148] A. S. Charles, A. Balavoine, and C. J. Rozell, "Dynamic filtering of time-varying sparse signals via l1 minimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 21, pp. 5644–5656, 2016.

[149] W. Gangbo, W. Li, S. Osher, and M. Puthawala, "Unnormalized optimal transport," *ArXiv preprint arXiv:1902.03367*, 2019.

[150] A. S. Charles, M. S. Asif, J. Romberg, and C. J. Rozell, "Sparsity penalties in dynamical system estimation," in *Proc of the CISS*, 2011, pp. 1–6.

[151] J. Wright, A. Ganesh, K. Min, and Y. Ma, "Compressive principal component pursuit," *Information and Inference: A Journal of the IMA*, vol. 2, no. 1, pp. 32–68, 2013.

[152] T. Bouwmans, N. S. Aybat, and E.-h. Zahzah, *Handbook of robust low-rank and sparse matrix decomposition: Applications in image and video processing*. Chapman and Hall/CRC, 2016.

[153] X. Ye, J. Yang, X. Sun, K. Li, C. Hou, and Y. Wang, "Foreground–background separation from video clips via motion-assisted matrix restoration," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 11, pp. 1721–1734, 2015.

[154] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 3, pp. 597–610, 2012.

[155] Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma, "Stable principal component pursuit," in *2010 IEEE international symposium on information theory*, IEEE, 2010, pp. 1518–1522.

[156] A. E. Waters, A. C. Sankaranarayanan, and R. Baraniuk, "Sparcs: Recovering low-rank and sparse matrices from compressive measurements," in *Advances in neural information processing systems*, 2011, pp. 1089–1097.

[157] W. C. Davidon, "Variable metric method for minimization," *SIAM Journal on Optimization*, vol. 1, no. 1, pp. 1–17, 1991.

[158] H. Janati, M. Cuturi, and A. Gramfort, "Wasserstein regularization for sparse multi-task regression," *ArXiv preprint arXiv:1805.07833*, 2018.

[159] A. Profeta and K.-T. Sturm, "Heat flow with dirichlet boundary conditions via optimal transport and gluing of metric measure spaces," *ArXiv preprint arXiv:1809.00936*, 2018.

[160] E. Mainini, "A description of transport cost for signed measures," *Journal of Mathematical Sciences*, vol. 181, no. 6, pp. 837–855, 2012.

[161] M. Cuturi and G. Peyré, "A smoothed dual approach for variational wasserstein problems," *SIAM Journal on Imaging Sciences*, vol. 9, no. 1, pp. 320–343, 2016.

[162] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[163] Y. Chen and X. Ye, "Projection onto a simplex," *ArXiv preprint arXiv:1101.6081*, 2011.

[164] H. Ltaief, S. Tomov, R. Nath, P. Du, and J. Dongarra, "A scalable high performant cholesky factorization for multicore with gpu accelerators," in *International Conference on High Performance Computing for Computational Science*, Springer, 2010, pp. 93–101.

[165] C. Cao, J. Dongarra, P. Du, M. Gates, P. Luszczek, and S. Tomov, "Clmagma: High performance dense linear algebra with opencl," in *Proceedings of the International Workshop on OpenCL 2013 & 2014*, ACM, 2014, p. 1.

[166] J. Dongarra, M. Gates, A. Haidar, Y. Jia, K. Kabir, P. Luszczek, and S. Tomov, "Portable hpc programming on intel many-integrated-core hardware with magma port to xeon phi," in *International Conference on Parallel Processing and Applied Mathematics*, Springer, 2013, pp. 571–581.

[167] S. Tomov, J. Dongarra, and M. Baboulin, "Towards dense linear algebra for hybrid gpu accelerated manycore systems," *Parallel Computing*, vol. 36, no. 5-6, pp. 232–240, 2010.

[168] V. Menon and K. Pingali, "Look left, look right, look left again: An application of fractal symbolic analysis to linear algebra code restructuring," *International Journal of Parallel Programming*, vol. 32, no. 6, pp. 501–523, 2004.

[169] A. M. Dale and M. I. Sereno, "Improved localizadon of cortical activity by combining eeg and meg with mri cortical surface reconstruction: A linear approach," *Journal of cognitive neuroscience*, vol. 5, no. 2, pp. 162–176, 1993.

[170] K. Uutela, M Hämäläinen, and E. Somersalo, "Visualization of magnetoencephalographic data using minimum current estimates," *NeuroImage*, vol. 10, no. 2, pp. 173–180, 1999.

[171] R. L. Buckner, J. R. Andrews-Hanna, and D. L. Schacter, "The brain's default network," *Annals of the New York Academy of Sciences*, vol. 1124, no. 1, pp. 1–38, 2008.

[172] C. Lamus Garcia Herreros, "Models and algorithms of brain connectivity, spatial sparsity, and temporal dynamics for the meg/eeg inverse problem," PhD thesis, Massachusetts Institute of Technology, 2015.

[173] H. Janati, T. Bazeille, B. Thirion, M. Cuturi, and A. Gramfort, "Group level meg/eeg source imaging via optimal transport: Minimum wasserstein estimates," *ArXiv preprint arXiv:1902.04812*, 2019.

[174] M. S. Hamalainen and J. Sarvas, "Feasibility of the homogeneous head model in the interpretation of neuromagnetic fields," *Physics in medicine & biology*, vol. 32, no. 1, p. 91, 1987.

[175] J.-L. Starck, D. L. Donoho, and E. J. Candès, "Astronomical image representation by the curvelet transform," *Astronomy & Astrophysics*, vol. 398, no. 2, pp. 785–800, 2003.

[176] G. Kutyniok and W.-Q. Lim, "Image separation using wavelets and shearlets," in *International Conference on Curves and Surfaces*, Springer, 2010, pp. 416–430.

[177] Y.-H. Yang, "On sparse and low-rank matrix decomposition for singing voice separation," in *Proceedings of the 20th ACM international conference on Multimedia*, ACM, 2012, pp. 757–760.

[178] S. A. Marley, C. Erbe, and C. P. S. Kent, "Underwater recordings of the whistles of bottlenose dolphins in fremantle inner harbour, western australia," *Scientific data*, vol. 4, p. 170 126, 2017.

[179] C. Ioana, A. Quinquis, and Y. Stephan, "Feature extraction from underwater signals using time-frequency warping operators," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 3, pp. 628–645, 2006.

[180] B. D. Charlton, J. L. Keating, L. Rengui, Y. Huang, and R. R. Swaisgood, "Female giant panda (ailuropoda melanoleuca) chirps advertise the caller's fertile phase," *Proceedings of the Royal Society B: Biological Sciences*, vol. 277, no. 1684, pp. 1101–1106, 2009.

[181] G Lopez-Risueno, J Grajal, and O Yeste-Ojeda, "Atomic decomposition-based radar complex signal interception," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 150, no. 4, pp. 323–331, 2003.

[182] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[183] E. K. Ryu, A. B. Taylor, C. Bergeling, and P. Giselsson, "Operator splitting performance estimation: Tight contraction factors and optimal parameter selection," *ArXiv preprint arXiv:1812.00146*, 2018.

[184] Y. Nesterov, "Gradient methods for minimizing composite functions," *Mathematical Programming*, vol. 140, no. 1, pp. 125–161, 2013.

[185] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[186] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, p. 9, 2016.

[187] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 114–141, 2003.

[188] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.

[189] G. K. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin, "Registration of 3d point clouds and meshes: A survey from rigid to nonrigid," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 7, pp. 1199–1217, 2013.

[190] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching," *Proceedings of the National Academy of Sciences*, vol. 103, no. 5, pp. 1168–1172, 2006.

[191] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, and M. Ovsjanikov, "Shape google: Geometric words and expressions for invariant shape retrieval," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 1, p. 1, 2011.

[192] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, "Functional maps: A flexible representation of maps between shapes," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 30, 2012.

[193] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[194] X. Shi, Q. Liu, W. Fan, S. Y. Philip, and R. Zhu, "Transfer learning on heterogenous feature spaces via spectral transformation," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, IEEE, 2010, pp. 1049–1054.

[195] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 361–368.

[196] Y. Han, F. Wu, D. Tao, J. Shao, Y. Zhuang, and J. Jiang, "Sparse unsupervised dimensionality reduction for multiple view data," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 10, p. 1485, 2012.

[197] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Advances in neural information processing systems*, 2008, pp. 1433–1440.

[198] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[199] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Unsupervised domain adaptation by domain invariant projection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 769–776.

[200] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1410–1417.

[201] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf, "Domain adaptation with conditional transferable components," in *International conference on machine learning*, 2016, pp. 2839–2848.

[202] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 2011, pp. 999–1006.

[203] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, pp. 2066–2073.

[204] C. Wang and S. Mahadevan, "Manifold alignment using procrustes analysis," in *Proceedings of the 25th international conference on Machine learning*, ACM, 2008, pp. 1120–1127.

[205] ——, "A general framework for manifold alignment," in *2009 AAAI Fall Symposium Series*, 2009.

[206] Z. Cui, H. Chang, S. Shan, and X. Chen, "Generalized unsupervised manifold alignment," in *Advances in Neural Information Processing Systems*, 2014, pp. 2429–2437.

[207] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5302–5316, 2009.

[208] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2960–2967.

[209] B. Sun and K. Saenko, "Subspace distribution alignment for unsupervised domain adaptation.," in *BMVC*, vol. 4, 2015, pp. 24–1.

[210] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[211] K. Thopalli, R. Anirudh, J. J. Thiagarajan, and P. Turaga, "Multiple subspace alignment improves domain adaptation," *ArXiv preprint arXiv:1811.04491*, 2018.

[212] L. V. Kantorovich, "On a problem of monge," *Journal of Mathematical Sciences*, vol. 133, no. 4, pp. 1383–1383, 2006.

[213] E. Arias-Castro, A. Javanmard, and B. Pelletier, "Perturbation bounds for procrustes, classical scaling, and trilateration, with applications to manifold learning," *ArXiv preprint arXiv:1810.09569*, 2018.

[214] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, International Society for Optics and Photonics, vol. 1611, 1992, pp. 586–607.

[215] C. Pandarinath, K. C. Ames, A. A. Russo, A. Farshchian, L. E. Miller, E. L. Dyer, and J. C. Kao, "Latent factors and dynamics in motor cortex and their application to brain–machine interfaces," *Journal of Neuroscience*, vol. 38, no. 44, pp. 9390–9401, 2018.

[216] E. L. Dyer, M. G. Azar, M. G. Perich, H. L. Fernandes, S. Naufel, L. E. Miller, and K. P. Körding, "A cryptography-based approach for movement decoding," *Nature Biomedical Engineering*, vol. 1, no. 12, p. 967, 2017.

[217] B. Culpepper and B. A. Olshausen, "Learning transport operators for image manifolds," in *Advances in neural information processing systems*, 2009, pp. 423–431.

[218] N. Papadakis, "Optimal transport for image processing," PhD thesis, Université de Bordeaux, 2015.

[219] E. C. Chi and K. Lange, "Splitting methods for convex clustering," *Journal of Computational and Graphical Statistics*, vol. 24, no. 4, pp. 994–1013, 2015.

[220] J. Von Neumann, "A certain zero-sum two-person game equivalent to the optimal assignment problem," *Contributions to the Theory of Games*, vol. 2, no. 5-12, p. 7, 1953.